

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

JCES3 U.S. PRO
09/740011
12/20/00

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日
Date of Application:

2000年 6月 8日

出願番号
Application Number:

特願2000-177123

出願人
Applicant(s):

株式会社日立製作所

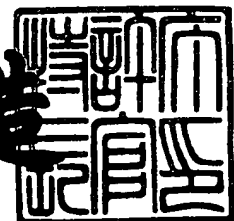
U.S. Appln. Filed 12-20-00
Inventor. K. Serizawa et al
Mattingly Stanger Malor
Docket NIT-245

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年11月10日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2000-3092863

【書類名】 特許願

【整理番号】 NT00P0190

【提出日】 平成12年 6月 8日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 13/38

【発明者】

 【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

 【氏名】 芹沢 一

【発明者】

 【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

 【氏名】 長須賀 弘文

【発明者】

 【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

 【氏名】 櫻庭 健年

【発明者】

 【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

 【氏名】 二瀬 健太

【発明者】

 【住所又は居所】 神奈川県秦野市堀山下 1 番地 株式会社日立製作所 エンタープライズサーバ事業部内

 【氏名】 山下 正弘

【特許出願人】

 【識別番号】 000005108

 【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100068504

【弁理士】

【氏名又は名称】 小川 勝男

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100086656

【弁理士】

【氏名又は名称】 田中 恭助

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100094352

【弁理士】

【氏名又は名称】 佐々木 孝

【電話番号】 03-3661-0071

【手数料の表示】

【予納台帳番号】 081423

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 計算機システムおよびそのデータ転送方法

【特許請求の範囲】

【請求項 1】 第 1 の計算機ノードとそれに接続された第 2 の計算機ノードを有する計算機システムであって、第 1 の計算機システムにあって、データレコードを格納する第 1 の記憶領域と、任意の時間間隔で第 2 の計算機ノードとは非同期に第 1 の記憶領域にデータレコードを格納する第 1 の処理部と、第 2 の計算機ノードにあって第 1 の記憶領域から転送されたデータレコードを記憶する第 2 の記憶領域と、第 1 の計算機ノードとは非同期の任意の時間間隔で第 1 の記憶領域から読み出すべきレコード群を指定して第 2 の記憶領域に読み出し、参照する第 2 の処理部を備えたことを特徴とする計算機システム。

【請求項 2】 第 2 の計算機ノードは第 2 の処理部を一定の時間間隔で起動し、第 1 の記憶領域から第 2 の記憶領域へのデータの読み出しを行なわしめるタイマを備えたことを特徴とする請求項 1 記載の計算機システム。

【請求項 3】 第 1 の処理部はデータレコードが格納された順序を示す識別番号を付与して前記データレコードを第 1 の記憶領域に格納し、第 1 の記憶領域は前記識別番号とデータレコードが組で格納される複数個のエントリを有しており、前記エントリに第 1 の処理部で書き込まれる方向と前記エントリから読み出される方向は逆であり、第 2 の処理部は第 2 の記憶領域に転送された第 1 の記憶領域のデータを参照し、前記識別番号によって当該データレコードが正しいものであるか否かを判断することを特徴とする請求項 1 記載の計算機システム。

【請求項 4】 第 1 の処理部では前記データレコードを書き込んでから当該データレコードの識別番号を書き込み、第 2 の処理部では第 2 の記憶領域に読み出されたデータの識別番号に連続性がある場合当該データレコードは正しいと判断し、連続性がない場合当該データレコードが不正であると判断することを特徴とする請求項 3 記載の計算機システム。

【請求項 5】 第 1 の処理部は更に前記データレコードに対して誤り検出符号を生成する誤り検出符号生成処理部を含み、第 1 の記憶領域に前記データレコードと前記誤り検出符号を書き込み、第 2 の処理部は第 2 の記憶領域に読み出された

データについて前記誤り検出符号により誤りの検出を行ない、誤りが検出されない場合当該データレコードを正しいものと判断し誤りが検出された場合当該データレコードを正しくないものと判断することを特徴とする請求項 1 記載の計算機システム。

【請求項 6】第 1 の記憶領域は前記誤り検出符号とデータレコードが組で格納される複数のエントリを有しており、前記エントリに第 1 の処理部で書き込まれる方向と前記エントリから読み出される方向とは同一方向であることを特徴とする請求項 5 記載の計算機システム。

【請求項 7】第 1 の計算機ノードと第 2 の計算機ノードが接続された計算機システムであり、第 1 の計算機ノードはデータレコードを格納する第 1 の記憶領域と、任意の時間間隔で第 2 の計算機ノードと非同期に第 1 の記憶領域に前記データレコードを格納する第 1 の処理部と、任意の時間間隔で第 1 の記憶領域の前記データレコードを第 2 の計算機ノードに送信するデータ送信要求を生成するデータ送信生成処理部とを有し、第 2 の計算機ノードは転送された第 1 の記憶領域のデータレコードを格納する第 2 の記憶領域と、第 2 の記憶領域のデータレコードを任意の時間間隔で第 1 の計算機ノードと非同期に参照する第 2 の処理部を備えたことを特徴とする計算機システム。

【請求項 8】第 1 の計算機ノード上の第 1 の記憶領域に格納された一つ以上のレコードで構成されるデータを、第 2 の計算機ノード上で動作しているプログラムが、当該記憶領域を指定して、第 2 の計算機ノード上の第 2 の記憶領域に転送せしめて参照することが可能な通信手段を有する計算機システムのデータ転送方法において、第 1 の計算機ノード上で動作し、任意の時間間隔で、第 1 の記憶領域に、一つ以上のレコードで構成されるデータを格納するステップと、第 2 の計算機ノード上で動作し、任意の時間間隔で、前記通信手段を用いて第 1 の記憶領域の指定されたデータを第 2 の記憶領域に転送せしめ、そのデータを参照するステップとを備えたことを特徴とするデータ転送方法。

【請求項 9】第 1 の記憶領域を持つ第 1 の計算機ノード上で動作しているプログラムが、第 2 の計算機ノードの主記憶装置内の第 2 の記憶領域に、第 1 の記憶領域の一つ以上のレコードで構成されるデータを、直接格納することが可能な通

信手段を有する計算機システムのデータ転送方法において、第 1 の計算機ノード上で動作し、任意の時間間隔で、前記通信手段を用いて第 2 の記憶領域に一つ以上のレコードで構成されるデータを格納するステップと、第 2 の計算機ノード上で動作し、任意の時間間隔で、第 2 の領域のデータを参照するステップとを備えたことを特徴とするデータ転送方法。

【請求項 1 0】第 1 の記憶領域は識別番号とデータレコードが組で格納される複数個のエントリを有しており、第 1 の計算機ノード上で動作し、前記データレコードを書き込んでから当該データレコードの前記識別番号を書き込み、前記エントリに書き込まれる方向と逆の方向に前記エントリから読み出すステップと、第 2 の計算機ノード上で動作し、第 2 の記憶領域に転送された第 1 の記憶領域のデータを参照し、第 2 の記憶領域に読み出されたデータの前記識別番号に連続性がある場合当該データレコードは正しいと判断し、連続性がない場合当該データレコードが不正であると判断するステップとを備えたことを特徴とする請求項 8 または 9 記載のデータ転送方法。

【請求項 1 1】第 1 の記憶領域は誤り検出符号とデータレコードが組で格納される複数個のエントリを有しており、第 1 の計算機ノード上で動作し、第 1 の記憶領域に前記データレコードとその誤り検出符号を書き込み、前記エントリに書き込まれる方向と同一方向に前記エントリから読み出すステップと、第 2 の計算機ノード上で動作し、第 2 の記憶領域に読み出されたデータについて前記誤り検出符号により誤りの検出を行ない、誤りが検出されない場合当該データレコードを正しいものと判断し誤りが検出された場合当該データレコードを正しくないものと判断するステップとを備えたことを特徴とする請求項 8 または 9 記載のデータ転送方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、ネットワークまたは入出力チャネルで接続された計算機の間で、複数のレコードからなるデータを受け渡す処理に関する。

【 0 0 0 2 】

【従来の技術】

二つの計算機ノード間で、多数のレコードからなるデータを受け渡す従来の技術として、以下に示すものがある。

【 0 0 0 3 】

第 1 の従来の技術は、特開平 6 - 6 7 9 4 4 号公報の 2 ページ 6 1 行目から、2 ページ 8 1 行目に示されている。この第 1 の従来の技術は、2 つの計算機ノード間で共用しているディスク装置を利用した方法である。この方法では、同一のデータを格納した 2 個のボリュームにからなる組を用意し、各々のボリュームを各計算機ノードに接続させて、共用できる状態にしておく。そして、一方の計算機ノードがデータを参照する場合は、ボリュームの組を解き（ボリュームの切り離し）、一方のボリューム（以下、第 1 のボリューム）を参照する側の計算機ノードに占有させる。その間、該ディスクの制御装置は、もう片方のボリューム（以下、第 2 のボリューム）に他方の計算機ノードによる変更を全て記録しておく。データを参照した計算機ノードが、参照を終え、上記第 1 のボリュームの占有を解くと、上記ディスクの制御装置は、上記第 2 のボリュームに対する変更の記録を、上記第 1 のボリュームに反映し、その後、上記 2 個のボリュームを、同一のデータを格納する組として、二つの計算機ノードから共用できる状態にする（ボリュームの再同期化）。

【 0 0 0 4 】

第 2 の従来の技術は、特開平 6 - 1 4 9 4 8 5 号公報の 3 ページ 5 8 行目から、4 ページ 5 2 行目に示されている。この第 2 の従来の技術は、計算機ノード間で共用している半導体外部記憶装置を利用した方法である。この方法では、単一のメモリ領域を複数の計算機ノード間で共用し、各々の計算機ノードは、該メモリ領域に対して排他的にアクセスする。

【 0 0 0 5 】

【発明が解決しようとする課題】

ここで、第 1 の従来の技術は、一方の計算機ノードがデータを参照するたびにボリュームの切り離しと、再同期化を行う必要が有る。そのため、リアルタイム

処理に適用することは困難である問題がある。

【0006】

一方、第2の従来の技術は、データ転送のたびに、レコードの完全性を保証するためにデータの出力側の計算機ノードと、データの参照側計算機ノードとの間で、これらの領域の排他制御を行う必要がある。大量のデータを転送する際には、これらの排他処理に要するCPUオーバーヘッドが膨大となる問題がある。さらに、このCPUオーバーヘッドは、データ転送の効率を低下させることがある。

【0007】

本発明の第1の目的は、データ転送効率の向上を阻害する排他処理に要するCPUオーバーヘッドを軽減することにある。

【0008】

本発明の第2の目的は、リアルタイム処理にも利用可能なデータ転送方法を提供することにある。

【0009】

【課題を解決するための手段】

本発明では、RDMA (Remote Direct Memory Access) を用いた、データ転送を行う。RDMAとは、送信側の計算機ノードの送信すべきデータのアドレスを受信側の計算機ノードが判っている、または受信側の計算機ノードの受信すべきデータのアドレスを送信側の計算機ノードがわかっているものである。そして、ネットワークで接続された二つの計算機ノード間において、一方の計算機ノードのプログラムが、該計算機ノードの主記憶のデータを格納すべき／データを読み出すべき領域と、他方の計算機ノードの主記憶のデータを読み出すべき／データを格納すべき領域とを指定してそれらの領域間でデータのコピーをする要求を作成し、この要求を、通信手段またはそれを制御するソフトウェアで処理することで、上記計算機ノードのそれぞれの主記憶間で直接データのコピーを行う技術である。

【0010】

RDMAには、RDMAを起動する計算機ノードの主記憶のデータを、他方の計算機ノード上の主記憶に格納するRDMA-Writeと、RDMAを起動す

る計算機ノードの主記憶に、他方の計算機ノード上の主記憶のデータを格納する RDMA-Read の 2 種がある。

【0011】

RDMA は、例えば米 Intel 社、米 Compaq 社、米マイクロソフト社による、Virtual Interface Architecture Specification 1.0 (1997.12.16) に記載されている。

【0012】

本発明では、第 1 の計算機ノードの主記憶上の領域に、同期を取らずに一方的に（非同期の手順で）任意の時間間隔で、一つ以上のレコードを格納し、第 2 の計算機ノード上で動作しているプログラムが、RDMA-Read を利用して、当該領域を任意の時間間隔で参照することで、データ転送を実現する。

【0013】

また、第 1 の計算機ノード上のプログラムが、第 2 の計算機ノードの主記憶上の領域に、同期を取らずに一方的に（非同期の手順で）任意の時間間隔で、RDMA-Write を用いて一つ以上のレコードを格納し、第 2 の計算機ノード上で動作しているプログラムが、上記領域を任意の時間間隔で参照することでデータ転送を実現する。

【0014】

【発明の実施の形態】

以下、本発明の実施の形態を、図を用いて説明する。

まず、図 1 から図 10 を用いて第 1 の本発明の実施の形態を説明する。

図 1 は、第 1 の本発明の実施の形態の全体構成図である。第 1 の計算機ノード 10 と第 2 の計算機ノード 20 は、ネットワーク 30 に接続している。そして、第 1 の計算機 10 と第 2 の計算機 20 とは、ネットワーク 30 を介して互いに通信することが可能である。

【0015】

第 1 の計算機ノード 10 には、第 2 の計算機ノード 20 に送信すべきデータレコード 153 を出力する第 1 のプログラム 110 と主記憶 150 に格納され、デ

ータレコード153を格納するデータレコードテーブル151と、前記データレコード153をネットワーク30を経由して第2の計算機20へ送信する送信部170を有している。これは、第1のプログラムと独立したプログラム、又はハードウェアで構成されている。さらに、第1のプログラム110は、前記データレコード153を出力するデータレコード出力処理部111と、後述する識別情報152を出力する識別情報出力処理部112と、データレコード出力処理部111および識別情報出力処理部112の出力をデータレコードテーブル151に格納する格納処理部113で構成される。ここで、前記識別情報152とは、少なくとも前後に連続して格納された2個のデータレコード153を識別できる情報であり、例えばデータレコード153のそれぞれに対し採番される通し番号である。識別情報出力処理部112には該通し番号を生成するカウンタ116を有する。さらに、格納処理部113には、データレコードテーブル151における格納すべきエントリのインデクス（ある識別情報及びデータレコードをどのエントリに格納すべきかを示すもの）を格納するポインタ115を含む。

【0016】

また、データレコード出力処理部111とは例えば、OLTP（On Line Transaction Processing）がジャーナルデータをデータレコードテーブル151に書き込むものであり、この例ではデータレコード153はオンライン処理におけるジャーナルデータである。

【0017】

第2の計算機ノード20は、第1の計算機10が出力したデータレコード153を受信し、参照する第2のプログラム210と第1の計算機10のデータレコードテーブル151の、完全または不完全な複製である主記憶250上のデータレコードテーブル251と、前記データレコード153をネットワーク30を経由して第1の計算機から受信する受信部270を有している。さらに、第2のプログラム210は、タイマ211と、前記データレコード153の受信要求を生成するデータ受信要求生成処理部212と、データレコード参照処理部221を有する。なお、タイマ211は一定の時間間隔でデータ受信要求生成処理部212を起動するための処理であり、第2のプログラム210の外部に存在しても良

い。

【0018】

さらに、データレコード参照処理部221はデータレコードテーブル251において、どのエントリを参照すべきかを示すインデクスを格納するポインタ225と、読み出された前記識別情報252の妥当性を検証する為に使用するカウンタ226とを含む。

【0019】

なお、データレコード153を生成するプログラム、およびデータレコード253を参照してさらに別の処理を行うプログラムは、本発明とは直接関係ないので、本実施例では省略する。

【0020】

図2は第1の実施の形態において、第1のプログラム110の処理を示すフローチャートである。

まず、データレコード出力処理部111が1個のデータレコード153を出力し、格納処理部113に該データレコード153のデータレコードテーブル151への格納処理を依頼する(ステップ11a)。格納処理部113は、識別情報出力処理部112を起動する(ステップ11b)。識別情報出力処理部112は識別情報152を出力し、これを格納処理部113に返す(ステップ11c)。格納処理部113は、ステップ11aのデータレコード153とステップ11cの識別情報152とを組にして、ポインタ115が指すエントリに格納する(ステップ11g)。その後、格納処理部113は、ポインタ115をインクリメントし、最大値を超えた場合はラップ処理を行う(ステップ11h)。識別情報出力処理部112および格納処理部113の詳細については後述する。第1のプログラム110が複数のデータレコード153を出力する場合、上記ステップ11aからステップ11hを繰り返す。識別情報及びデータレコードは第2の計算機ノードとは関係なくデータレコードが発生する毎にデータレコードテーブル151へと格納される。

【0021】

図3は、第1の実施の形態において、第2のプログラム210の処理を示すフ

ローチャートである。

まず、第2のプログラム210はデータレコードテーブル251を初期化する（ステップ21a）。初期化後のデータレコードテーブル251は、後に図7で説明するため、ここでは説明を省略する。次にデータ受信要求生成処理部212が参照先としてデータレコードテーブル151を、受信先としてデータレコードテーブル251を、それぞれ指定したデータ受信要求を生成し、受信部270を起動する（ステップ21c）。即ち、このときにRDMA-Readの起動が行なわれる。ステップ21cにおいて、送信先および受信先には、データレコードテーブル151および251の全てのエントリ、または一部のエントリ群、のいずれを指定しても良い。望ましくは、前回の最後のデータ転送において、送信部170が最後のエントリを読み出した時刻から、今回のデータ転送において、送信部が最初のエントリを読み出す時刻までの間に、格納処理部113が格納するエントリ群を指定する。第1の計算機ノード10の負荷により、上記エントリ群に含まれるエントリ数が変化する場合、負荷に追従して読み出すエントリ数を増減する。例えば、前回読み出しに失敗したエントリ数が多い場合は、次に読み出すエントリ数を減少させる。

【0022】

さらに、第2のプログラム210は、受信部270からの、ステップ21cで発行したデータ転送の完了を待つ（ステップ21d、ステップ21e）。さらに、データレコード受信処理部221が、データレコードテーブル251を参照する（ステップ21f）。データレコード参照処理部221については、後に図10を用いて説明する。さらに、第2のプログラム210は、タイマ211に対し、一定時間後にステップ21cから処理を続行することを要求する（ステップ21g）。ステップ21gにおいて、タイマに要求する時間間隔は任意で良い。リアルタイム性を向上する為に望ましくは、該時間間隔として、今回のデータ転送において、送信部170が最後のエントリを読み出した時刻から、次回のデータ転送において、送信部170が最初のエントリを読み出す時刻までの間に、格納処理部113が一つ以上のエントリを格納可能な時間間隔を指定する。特に、ステップ21fにおいて、今回データ転送した全てのエントリが読み出し可能であ

った場合には、次のデータが既に格納されている可能性が有るので、時間間隔として0を設定することが望ましい。

【 0 0 2 3 】

データ転送効率を向上する為に望ましくは、該時間間隔として今回のデータ転送において、送信部170が最後のエントリを読み出した時刻から、次回のデータ転送において、送信部が最初のエントリを読み出す時刻までの間に、格納処理部113がデータレコードテーブル151の半分に相当するエントリを格納可能な時間間隔を指定する。次に、タイマ211が一定の時間後にデータ受信要求生成処理部212を起動する（ステップ21b）。このようにRDMAに関して第1の計算機ノードにおけるデータの格納と第2の計算機ノードにおけるRDMA-Readによるデータの読み出しがそれぞれ任意の時間間隔で非同期に行なわれるので、これらの間での確認手順が不要であり、プログラムにかかる負担は小さいものとなる。

【 0 0 2 4 】

以下では、1、m、nは1を超える自然数とし、1とn、およびmとnはそれぞれ互いに素であるとする。n-1はカウンタ116の上限値の意味を持ち、mはデータレコードテーブル151のエントリ数を、1はデータレコードテーブル251のエントリ数をそれぞれ示す。

【 0 0 2 5 】

図4は、第1の実施の形態において、識別情報出力処理部112の処理を示すフローチャートである。

まず識別情報出力処理部112は、カウンタ116を0にクリアし（ステップ112a）、格納処理部113からの要求を待つ（ステップ112b、ステップ112c）。ここで、格納処理部113からの要求があると、識別情報出力処理部112はカウンタ116の値を格納処理部113に返す（ステップ112d）。ここで、カウンタ116の値がn-1より小さいか判断し（ステップ112e）、該判断が真の場合、カウンタ116をインクリメントし（ステップ112f）、ステップ112bからの処理を繰り返す。該判断が偽の場合は、ステップ112aからの処理を繰り返す。

【 0 0 2 6 】

図 5 は、第 1 の実施の形態において、格納処理部 1 1 3 を示すフローチャートである。

まず格納処理部 1 1 3 は、データレコードテーブル 1 5 1 を初期化する（ステップ 1 1 3 a）。初期化後のデータレコードテーブル 1 5 1 は後に図 8 に説明する。さらに格納処理部 1 1 3 は、ポインタ 1 1 5 を 0 にクリアし（ステップ 1 1 3 b）、データレコード出力処理部 1 1 1 からの要求を待つ（ステップ 1 1 3 c、ステップ 1 1 3 d）。ここで、データレコード出力処理部 1 1 1 からの要求があると、格納処理部 1 1 3 は、データレコード出力処理部 1 1 1 が出力したデータレコード 1 5 3 を受け取り（ステップ 1 1 3 e）、さらに図 2 で示したステップ 1 1 b を行い、識別情報出力処理部 1 1 2 から識別情報 1 5 2 を得る（ステップ 1 1 3 f）。さらに、格納処理部 1 1 3 は、データレコードテーブル 1 5 1 においてポインタ 1 1 5 が指すエントリに、ステップ 1 1 3 e で得たデータレコード 1 5 3 を格納し（ステップ 1 1 3 g）、ステップ 1 1 3 f で得た識別情報 1 5 2 を、該エントリに格納する（ステップ 1 1 3 h）。さらにポインタのインクリメントのための処理を行う（ステップ 1 1 3 k、ステップ 1 1 3 l）。

【 0 0 2 7 】

図 6 は、第 1 の実施の形態において、格納処理部 1 1 3 のステップ 1 1 3 a の直後の、即ち初期化後のデータレコードテーブル 1 5 1 を示している。

【 0 0 2 8 】

データレコードテーブルはエントリ 0 からエントリ $m-1$ で成り立っており、それぞれポインタが 0 から $m-1$ をとるときに指すエントリに対応している。格納処理部 1 1 3 は各エントリの識別情報 1 5 2 を、次のように格納する。格納処理部 1 1 3 はエントリ 0 の識別情報 1 5 2. 0 に -1 を、エントリ 1 の識別情報 1 5 2. 1 からエントリ $m-2$ の識別情報 1 5 2. $m-2$ までは、0、1、2、... の順に、0 から 1 ずつ増加させた数を格納する。ここで、もし、識別情報 1 5 2 に格納すべき値が $n-1$ を超えた場合には、該エントリの識別情報 1 5 2 には 0 を格納し、以降は同様に 1 ずつ増加させた数を格納する。さらに格納処理部 1 1 3 は、エントリ $m-1$ の識別情報 1 5 2. $m-1$ には $n-1$ を格納する。

各エントリのデータレコード153は、適当な初期値で初期化されている。但し、以降に述べるデータレコード参照処理部221では、これらのデータレコード153を無視するため、必ずしも初期化する必要はない。

【0029】

図7は、本発明の第1の実施の形態において、第2のプログラム210のステップ21aの直後の、即ち初期化後のデータレコードテーブル251を示している。図6との違いは、エントリ数がmではなく1（エル）であることである。

【0030】

図8は、本発明の第1の実施の形態において、格納処理部113と送信部170がそれぞれデータレコードテーブル151に対する書き込みと読み出しを行っているときの、ある一時点の状態を示している。

【0031】

矢印156は格納処理部113がエントリを書き込む方向を、矢印157は送信部170がエントリを読み出す方向を示している。すなわち、格納処理部113と送信部170とは互いに逆の順序で読み書きしている。この理由は、格納処理部113の書き込みと、送信部170の読み出しとのすれ違いを、識別情報152の不連続により検出するためである。以下に詳しく説明する。

【0032】

格納処理部113は、エントリ9のデータレコード153. 9、エントリ9の識別情報152. 9、エントリ10のデータレコード153. 10、エントリ10の識別情報152. 10、... の順にデータレコードテーブル151のエントリを書き込んでおり、送信部170の読み出しとすれ違う瞬間は、エントリ12のデータレコード153. 12を書き換え途中にある。送信部170は、エントリ12の識別情報152. 12、エントリ12のデータレコード153. 12、エントリ11の識別情報152. 11、エントリ11のデータレコード152. 11、... の順にデータレコードテーブル151のエントリを読み出している。

【0033】

ここで、識別情報152. 12とデータレコード153. 12、および識別情

報 152. 11 に注目する。格納処理部 113 は、データレコード 153. 12 の書き込みを完了してから識別情報 152. 12 を書き込む。そのため、図 8 に示した時点では、識別情報 152. 12 に格納処理 113 が書き込む前の値（具体的には 12）が残っていて、この値は識別情報 152. 11 の値（具体的には 68）とは不連続になる。この時、送信部 170 は既に識別情報 152. 12 を読み込んでおり、その後識別情報 152. 11 を読み出す。そのため、上記のすれ違いが生じたときは、必ず識別番号 152. 11 と識別番号 152. 12 が不連続になる。不連続になるということはまだ、エントリ 12 は書き換え中のデータレコードを含んでいることを意味する。

【0034】

なお、もし仮に格納処理部 113 と送信部 170 とが同じ順序で読み書きした場合はこの限りではない。これを図 9 および図 10 を用いて説明する。図 9 では、読み出し動作が書き込み動作に追いついて来た場合を示している。格納処理部 113 が識別番号 152. 11 を書き込んで（615. 1）から、送信部 170 がこれを読み出す（615. 2）。その後、送信部 170 がデータレコード 153. 12 を読み出している（615. 3）が、このデータレコード 153. 12 は格納処理部 113 が書き込みを完了する前であるから、不正な値のまま読み出される。その後の処理は図 10 に示す。

【0035】

図 10 では、まず格納処理部 113 がデータレコード 153. 12 を書き込む（615. 4）。その後、格納処理部 113 が識別番号 152. 12 を書き込んで（615. 5）から送信部 170 がこれを読み出し（615. 6）ているが、この識別番号 152. 12 は正しい値（具体的には 69）が読み出される。すなわち、送信部 170 が読み出した識別番号 152. 11 と識別番号 152. 12 は、連続になる。このように、格納処理部 113 の書き込みと、送信部 170 の読み出しとの順序が同じであると、図 9 で示したように、69 番目のデータレコードは正しくないまま読み出しているにも係わらず識別番号は連続したものとして読み出されてしまう。従って、識別番号 152 の連続性だけでデータレコード 153 が正しく読めたことを保証できない。

【 0 0 3 6 】

図 1 1 は本発明の第 1 の実施の形態において、図 8 のデータレコードテーブル 1 5 1 が、送信部 1 7 0 と受信部 2 7 0 とによって、第 2 の計算機ノード 2 0 に転送された、データレコードテーブル 2 5 1 を示している。

【 0 0 3 7 】

図 8 で述べたように、エントリ 9 の識別情報 2 5 2. 9 から、エントリ 1 1 の識別情報 2 5 2. 1 1 は 6 6 から 6 8 と連続しており、これらに対するデータレコード 2 5 3. 9 からデータレコード 2 5 3. 1 1 が正しく書き込まれたことを示している。さらに、データレコード 2 5 3. 1 2 の識別情報は 1 2 であり連続でなく、データレコード 2 5 3. 1 2 が書き換え途中に読み出されたことを示している。

【 0 0 3 8 】

図 1 2 は、第 1 の実施の形態において、データレコード参照処理部 2 2 1 の内容を示すフローチャートである。ここでは、識別番号がエントリ上で連続性があるか、どのエントリのデータレコードが不正であるかを知る。

【 0 0 3 9 】

まず、データレコード参照処理部 2 2 1 はカウンタ 2 2 6 を 0 にクリアし（ステップ 2 2 1 a）、ポインタ 2 2 5 を 0 にクリアする（ステップ 2 2 1 b）。ここで、データレコード参照処理部 2 2 1 はポインタ 2 2 5 が 0 であるか否かの判断を行い（ステップ 2 2 1 c）、該判断が真の場合は、ステップ 2 2 1 d へ、該判断が偽の場合は、ステップ 2 2 1 h へ進む。ステップ 2 2 1 d では、データレコード参照処理部 2 2 1 は、カウンタ 2 2 6 が 0 であるか否かの判断を行い、該判断が真の場合は、ステップ 2 2 1 e へ、該判断が偽の場合は、ステップ 2 2 1 g へ進む。ステップ 2 2 1 e では、データレコード参照処理部 2 2 1 は、インデクス 1 - 1 の識別情報 2 5 2 が $n - 1$ であるか否かの判断を行い、該判断が真の場合はステップ 2 2 1 f へ、該判断が偽の場合はノード B へ進む。ステップ 2 2 1 f では、データレコード参照処理部 2 2 1 は、ポインタ 2 2 5 が指すエントリの識別情報 2 5 2 が、カウンタ 2 2 6 の値と一致しているか否かを判断する。該判断が真の場合はノード A へ進み、該判断が偽の場合はノード B へ進む。ステッ

ブ 2 2 1 g において、データレコード参照処理部 2 2 1 は、インデクス 1 - 1 が指す識別情報 2 5 2 の値がカウンタ 2 2 6 の値から 1 を減じた数と等しいか否かを判断する。該判断が真の場合はステップ 2 2 1 f へ進み、該判断が偽の場合はノード B へ進む。ステップ 2 2 1 h において、データレコード参照処理部 2 2 1 は、カウンタ 2 2 6 が 0 か否かの判断を行う。該判断が真の場合、ステップ 2 2 1 i へ進み、該判断が偽の場合、ステップ 2 2 1 j へ進む。ステップ 2 2 1 i において、データレコード参照処理部 2 2 1 は、ポインタから 1 を減じたインデクスの、識別情報 2 5 2 の値が、 $n - 1$ であるか否かの判断を行う。該判断が真の場合はステップ 2 2 1 f へ進み、該判断が偽の場合はノード B へ進む。ステップ 2 2 1 j において、データレコード参照処理部 2 2 1 は、ポインタから 1 を減じたインデクスの、識別情報 2 5 2 の値が、カウンタ 2 2 6 の値から 1 を減じた数と等しいか否かの判断を行う。該判断が真の場合はステップ 2 2 1 f へ進み、該判断が偽の場合はノード B へ進む。ノード A に進んだ場合は、データレコード参照処理部 2 2 1 は、ポインタ 2 2 5 が指すエントリのデータレコード 2 5 3 は正しいと判断し、該データレコード 2 5 3 を参照する（ステップ 2 2 1 k）。さらにデータレコード参照処理部 2 2 1 は、カウンタ 2 2 6 が $n - 1$ より小さいか判断する。該判断が真の場合は、カウンタ 2 2 6 をインクリメントする（ステップ 2 2 1 n）。該判断が偽の場合はカウンタ 2 2 6 を 0 にクリアする（ステップ 2 2 1 o）。さらに、データレコード参照処理部 2 2 1 は、ポインタ 2 2 5 が 1 - 1 より小さいか否かを判断する。該判断が真の場合は、データレコード参照処理部 2 2 1 は、ポインタ 2 2 5 をインクリメントして（ステップ 2 2 1 q）、ステップ 2 2 1 c へ進む。該判断が偽の場合は、ステップ 2 2 1 b へ進む。ノード B へ進んだ場合は、データレコード参照処理部 2 2 1 は、ポインタ 2 2 5 が指すエントリのデータレコード 2 5 3 は不正と判断し、該データレコード 2 5 3 は参照しない（ステップ 2 2 1 l）。さらに、データレコード参照処理部 2 2 1 は、ポインタ 2 2 5、カウンタ 2 2 6 の値を変更せずに、ステップ 2 2 1 c から処理を繰り返す。

【 0 0 4 0 】

ここで、データレコードが不正と判断された場合はポインタはインクリメント

されない。従って、次のデータの転送はこのポインタで表されるエントリを含むいくつかのエントリのデータレコードが適当な時間間隔後に転送されることになる。

【 0 0 4 1 】

本発明の第 1 の実施形態では CPU の負荷の削減効果が大きい。

【 0 0 4 2 】

次に、図 1 3 から図 1 8 を用いて、第 2 の本発明の実施の形態と第 1 の本発明の実施の形態との違いを説明する。

図 1 3 は、第 2 の本発明の実施の形態の全体構成図である。図 1 との違いは、第 1 のプログラム 1 1 0 が誤り検出符号生成処理部 1 1 4 を含むことと、第 2 のプログラム 2 1 0 が誤り検出符号検査処理部 2 2 2 を含むこと、データレコードテーブル 1 5 1 に誤り検査符号 1 5 4 を含むこと、データレコードテーブル 2 5 1 に誤り検査符号 2 5 4 を含むことである。

【 0 0 4 3 】

誤り検出符号生成処理部 1 1 4 は、データレコードエントリの識別情報 1 5 2 と、データレコード 1 5 3 の組から、誤り検出符号 1 5 4 を生成する。誤り検出符号検査処理部 2 2 2 は、誤り検出符号 2 5 4 が、データレコードエントリの識別情報 2 5 2 と、データレコード 2 5 3 との組から生成された符号（誤りなし）か否（誤りあり）かを検査する。ここで、誤り検出符号を採用した理由を簡単に説明しておく。第 1 の実施例ではデータレコードテーブル 1 5 1 にジャーナルデータなどのデータレコードを書き込む方向と反対の方向に転送のための読み出しを進めることにより不正なデータを検出する方法を採ったのに対して、同一方向に読み出す場合を想定している。そして、データレコードの正しさを誤り検出符号を利用して保証しようとしたものである。

【 0 0 4 4 】

図 1 4 は、第 2 の本発明の実施の形態において、第 1 のプログラム 1 1 0 の処理を示すフローチャートである。

図 2 との違いは、ステップ 1 1 d およびステップ 1 1 e、ステップ 1 1 f を含むこと、ステップ g を含まないことである。ステップ 1 1 d では、格納処理部 1

1 3 がステップ 1 1 a のデータレコード 1 5 3 とステップ 1 1 c の識別情報 1 5 2 とを組にして、誤り検出符号生成処理部 1 1 4 に渡す。ステップ 1 1 e では、誤り検出符号生成処理部 1 1 4 が、ステップ 1 1 d で渡された情報から誤り検出符号 1 5 4 を生成し、格納処理部 1 1 3 に返す。ステップ 1 1 f では、格納処理部 1 1 3 が、ステップ 1 1 a のデータレコード 1 5 3 とステップ 1 1 c の識別情報 1 5 2 とステップ 1 1 e の誤り検出符号 1 5 4 とを組にして、ポインタ 1 1 5 が指すエントリに格納する。

【 0 0 4 5 】

図 1 5 は、第 2 の本発明の実施の形態において、格納処理部 1 1 3 の内容を示すフローチャートである。

図 5 との違いは、ステップ 1 1 3 i とステップ 1 1 3 j を含むことである。ステップ 1 1 3 i において、格納処理部 1 1 3 は、図 1 4 で示したステップ 1 1 d を行い、誤り検出符号生成処理部 1 1 4 から誤り検出符号 1 5 4 を得る。ステップ 1 1 3 j において、格納処理部 1 1 3 は、ステップ 1 1 3 i で取得した誤り検出符号 1 5 4 をポインタ 1 1 5 が指すエントリに格納する。

【 0 0 4 6 】

図 1 6 は、本発明の第 2 の実施の形態において、初期化後のデータレコードテーブル 1 5 1 を示している。

図 6 との違いは、誤り検出符号 1 5 4 を含むこと、データレコード 1 5 3 の初期化を必ず行う必要があることである。何故ならば古いデータではあるがデータレコードと誤り検出符号とが整合したデータが残っているとそれが新しく格納された正しいデータレコードなのか、古い（正しくない）データなのかの区別が付かなくなるからである。格納処理部 1 1 3 は、ステップ 1 1 3 a において、これらの誤り検出符号 1 5 4 を、識別情報 1 5 2、データレコード 1 5 3 から生成した誤り訂正符号以外の値、即ち、不正な符号を格納する。

【 0 0 4 7 】

図 1 7 は、本発明の第 2 の本発明の実施の形態において、データレコードテーブル 1 5 1 の、ある一時点の状態を示している。

図 8 との違いは、誤り検出符号 1 5 4 を含むことである。図 1 7 で、格納処理

部 1 1 3 が エントリ 1 2 の データ レコード 1 5 3 に 書き 込 ん で い る 途 中 で、 送 信 部 1 7 0 が 当 該 エントリ を 読 み 出 し て い る と す る。 こ の と き、 エントリ 1 2 の 誤 り 検 出 符 号 1 5 4 . 1 2 は、 当 該 エントリ から 生 成 さ れ た 誤 り 検 出 符 号 で な い た め、 不 正 な 符 号 と な る。 図 1 7 の そ の 他 の 誤 り 検 出 符 号 1 5 4 . 9 か ら 誤 り 検 出 符 号 1 5 4 . 1 1 は そ れ ぞ れ の エントリ から 生 成 さ れ た 誤 り 検 出 符 号 で あ る た め、 正 し い 符 号 で あ る。

【 0 0 4 8 】

図 1 8 は、 本 発 明 の 第 2 の 実 施 の 形 態 に お い て、 データ レコード 参 照 処 理 部 2 2 1 の 内 容 を 示 す フローチャート である。 こ こ で は 誤 り 検 出 符 号 に よ っ て 読 み 出 さ れ た データ レコード が 正 し い も の か ど う か を 判 定 し て い る。

【 0 0 4 9 】

図 1 2 と の 違 い は、 ステ ッ プ 2 2 1 r お よ び 2 2 1 s を 含 む こ と、 ステ ッ プ 2 2 1 c か ら ステ ッ プ 2 2 1 e、 お よ び ステ ッ プ 2 2 1 g か ら ステ ッ プ 2 2 1 j を 含 ま な い こ と で あ る。 ステ ッ プ 2 2 1 r に お い て、 データ レコード 参 照 処 理 部 2 2 1 は ポ イ ン タ 2 2 5 が 指 す エントリ を 誤 り 検 出 符 号 検 査 処 理 部 2 2 2 に 渡 す。 ステ ッ プ 2 2 1 s に お い て、 データ レコード 参 照 処 理 部 2 2 1 は、 誤 り 検 出 符 号 検 査 処 理 部 2 2 2 の 結 果 か ら、 当 該 レコード に 誤 り が 含 ま れ た か 否 か を 判 断 す る。 該 判 断 が 真 の 場 合 は ノード B へ 進 み、 該 判 断 が 偽 の 場 合 は ノード A に 進 む。

【 0 0 5 0 】

本 発 明 の 第 2 の 実 施 の 形 態 に お い て は、 読 み 取 り 順 序 が 同 一 方 向 の も の で も データ レコード の 正 し さ が 保 証 で き る。

【 0 0 5 1 】

次 に、 図 1 9 か ら 図 2 2 を 用 い て、 第 3 の 本 発 明 の 実 施 の 形 態 と 第 1 の 本 発 明 の 実 施 の 形 態 と の 違 い を 説 明 す る。

図 1 9 は、 第 3 の 本 発 明 の 実 施 の 形 態 の 全 体 構 成 図 である。 図 1 と の 違 い は、 第 1 の プログラム 1 1 0 が データ 送 信 要 求 生 成 処 理 部 1 2 2 を 含 む こ と、 第 2 の プログラム 2 1 0 が タイマ 2 1 1 と データ 受 信 要 求 生 成 処 理 部 2 1 2 を 含 ま な い こ と、 である。 データ 送 信 要 求 生 成 処 理 部 1 2 2 は 送 信 部 1 7 0 に 対 し、 データ レコード テーブル 1 5 1 の 送 信 要 求 を 生 成 す る。 即 ち、 R D M A - W r i t e を

使って送信側の主導によりデータの転送を行なう場合を示している。このとき、第 1 の計算機ノードには図 1 の構成に比べて負荷はかかるが同期を取らない転送方式としているため従来例よりも負荷は小さいものとなっている。データ送信要求生成処理部 1 2 2 は送信データがある程度たまったときに送信部 1 7 0 にデータの読み出しを行なわせるものである。データの転送は第 2 のプログラムがポーリングして転送データがあることを認知して実施される。

【 0 0 5 2 】

図 2 0 は、本発明の第 3 の実施の形態における、第 1 のプログラム 1 1 0 の処理を示すフローチャートである。

図 2 との違いは、ステップ 1 1 i からステップ 1 1 l を含むことである。ステップ 1 1 i において、第 1 のプログラム 1 1 0 はデータレコードテーブル 1 5 1 を送信するか否かの判断をする。即ち、本発明の第 3 の実施の形態において、データ送信の間隔は任意である。リアルタイム性を向上する為に望ましくは、該時間間隔を短くする、即ち、なるべく多くの場合についてステップ 1 1 i における判断を真とする。データ送信の時間間隔の調整はデータ送信要求生成処理部 1 2 2 がステップ 1 1 i で行なう。

【 0 0 5 3 】

データ転送効率を向上する為に望ましくは、最後にデータ送信を行った時点から現在までに、格納処理部 1 1 3 がデータレコードテーブル 1 5 1 の半分に相当するエントリを格納したときにステップ 1 1 i における判断を真とする。

【 0 0 5 4 】

該判断が真の場合、ステップ 1 1 j に進み、偽の場合は 1 1 a に進む。ステップ 1 1 j においては、データ送信要求生成処理部 1 2 2 が、データ送信要求を作成し、送信部 1 7 0 を起動する。さらに、ステップ 1 1 k とステップ 1 1 l において、第 1 のプログラム 1 1 0 は、ステップ 1 1 j で起動した送信が完了したことを待つ。その後、第 1 のプログラム 1 1 0 はステップ 1 1 a から処理を続行する。本発明の第 3 の実施の形態では、第 2 のプログラム 2 1 0 は、図 3 に示すステップ 2 1 f を実行するのみである。

【 0 0 5 5 】

本発明の第 3 の実施の形態では、格納処理部 1 1 3 は、図 5 に示す処理と同一だが、ステップ 1 1 2 g とステップ 1 1 3 h の順序の入れ替えが可能であることが異なる。本発明の第 3 の実施の形態では、データレコードテーブル 1 5 1 の読み書き順序について、図 8 に示す順序でなくとも構わない。

【 0 0 5 6 】

図 2 1 は、本発明の第 3 の実施の形態において、データレコードテーブル 1 5 1 が、送信部 1 7 0 と受信部 2 7 0 とによって、第 2 の計算機ノード 2 0 に転送された、データレコードテーブル 2 5 1 を示している。

【 0 0 5 7 】

矢印 2 5 6 は受信部 2 7 0 がエントリを書き込む方向を、矢印 2 5 7 は第 2 のプログラム 2 1 0 がエントリを読み出す方向を示している。データレコード 2 5 3 が正しく読めたか否かの判定は、本発明の第 1 の実施の形態と同様である。

【 0 0 5 8 】

図 2 2 は、第 3 の実施の形態において、データレコード参照処理部 2 2 1 の内容を示すフローチャートである。

図 2 2 は図 1 2 と異なり、ステップ 2 1 1 u からステップ 2 1 1 x を含み、ステップ 2 1 1 m からステップ 2 1 1 q を含まず、ステップ 2 1 1 f をステップ 2 2 1 c の前に実行する。即ち、本発明の第 3 の実施の形態によれば、データレコード参照処理部 2 2 1 はインデクスの大きいほうの識別情報 2 5 2 から参照する。ステップ 2 1 1 u では、データレコード参照処理部 2 1 1 はポインタ 2 2 5、カウンタ 2 2 6 それぞれに、同じ任意の自然数を加える。ここで望ましくは、該自然数は以降のステップ 2 2 1 f の判断が真となる、最大の数とする。ただし、この最適値を予測することは困難なので、例えば前回のステップ 2 2 1 u において加算した自然数を記憶しておき、今回のステップ 2 1 1 u では、該自然数から、前回のステップ 2 2 1 u 処理から不正レコードを読み出した数を差し引いた数、またはそれに近い数を指定する。

【 0 0 5 9 】

さらに、ステップ 2 1 1 v では、データレコード参照処理部 2 1 1 はポインタ

225、カウンタ226のラップ処理を行う。すなわち、ポインタ225が0以上1-1以下になるまで、ポインタ225から1を繰り返し除し、カウンタ226が0以上n-1以下になるまで、カウンタ226からnを繰り返し除す。ステップ211fの判断が真の場合、データレコード参照処理部211はステップ211cへ進み、偽の場合はノードBへ進む。ステップ221Wにおいて、データレコード参照処理部211はポインタ225から1を減じた値が指すエントリ（ポインタ225が0の場合は、エントリ1-1）が既に正しく読み込み済みか判断する。

【0060】

該判断が真の場合、データレコード参照処理部211はステップ221uに進み、該判断が偽の場合、ステップ211xに進む。ステップ211xにおいて、データレコード参照処理部211は、カウンタ226、ポインタ225をそれぞれデクリメントする。本発明の第3の実施の形態ではRDMA-Readをサポートしていないものにも適用できる。

【0061】

次に、図23から図24を用いて、第4の本発明の実施の形態と第2の本発明の実施の形態との違いを説明する。

図23は、第4の本発明の実施の形態の全体構成図である。図23は、図13と異なり、第1のプログラム110がデータ送信要求生成処理部122を含み、第2のプログラム210がタイマ211とデータ受信要求生成処理部212を含まない。データ送信要求生成処理部122は本発明の第3の実施の形態と変わらない。

【0062】

図24は、第4の本発明の実施の形態における、第1のプログラム110の処理を示すフローチャートである。

図24は、図14と異なり、ステップ11iからステップ11lを含む。これらのステップは、本発明の第3の実施の形態と同様である。本発明の第4の実施の形態では読み取り順序が同一で、RDMA-Readをサポートしていないものにも適用出来る。

【 0 0 6 3 】

次に、図 2 5 から図 2 7 を用いて、第 5 の本発明の実施の形態と第 1 の本発明の実施の形態との違いを説明する。図 2 5 は、第 5 の本発明の実施の形態の全体構成図である。図 2 5 は図 1 と異なり、第 1 のプログラム 1 1 0 に通知処理部 1 1 7 を含み、第 2 のプログラム 2 1 0 にタイマ 2 1 1 を含まない。通知処理部 1 1 7 はデータレコードがある程度たまったとき送信部 1 7 0 に読み出しをなさせるとともに第 2 のプログラムに読み取りのきっかけを与えるもので R D M A - W r i t e の割り込みをするものである。

【 0 0 6 4 】

図 2 6 は、第 5 の本発明の実施の形態における、第 1 のプログラム 1 1 0 の処理を示すフローチャートである。

図 2 6 は、図 2 と異なり、ステップ 1 1 m を有する。ステップ 1 1 m において、通知処理部 1 1 7 は、送信部 1 7 0 に、第 2 のプログラム 2 1 0 が受信するイベント通知を依頼する。本実施例では第 2 のプログラムにタイマが不要で送信側のトリガでデータの転送が制御される。つまり、データの転送の時間間隔は通知処理部 1 1 7 の制御による。

【 0 0 6 5 】

図 2 7 は、第 5 の実施の形態において、第 2 のプログラム 2 1 0 の処理を示すフローチャートである。

図 2 7 は、図 3 と異なり、ステップ 2 1 h を含み、ステップ 2 1 b およびステップ 2 1 g を含まない。ステップ 2 1 b において、第 2 のプログラム 2 1 0 は通知処理部 1 1 7 からのイベント通知を待つ。本発明の第 5 の実施の形態では C P U の負荷の削減効果が大きく、送信側の通知をトリガとしてデータの転送が行なわれるので受信側の負担が小さい。

【 0 0 6 6 】

次に、図 2 8 を用いて、第 6 の本発明の実施の形態と第 2 の本発明の実施の形態との違いを説明する。

図 2 8 は、第 6 の本発明の実施の形態の全体構成図である。

【 0 0 6 7 】

図 2 8 は図 1 3 と異なり、第 1 のプログラム 1 1 0 に通知処理部 1 1 7 を含み、第 2 のプログラム 2 1 0 にタイマ 2 1 1 を含まない。その他の、第 6 の本発明の実施の形態と第 2 の本発明の実施の形態との違いは、第 5 の本発明の実施の形態と第 1 の本発明の実施の形態との違いと同一なので説明を省略する。これはデータレコードテーブル 1 5 1 へのジャーナルデータなどのデータレコードを書き込む方向とその読み出し方向とが同じである場合を想定し、且つ、送信側でデータの転送時間間隔を制御する場合である。本発明の第 6 の実施の形態では読み取り順序が同一のものでもサポートでき、CPU 負荷の削減の効果が大きい。また、送信側の通知をトリガとしているので受信側の負担が小さい。

【 0 0 6 8 】

次に、図 2 9 を用いて、第 7 の本発明の実施の形態と第 3 の本発明の実施の形態との違いを説明する。

図 2 9 は、第 7 の本発明の実施の形態の全体構成図である。図 2 9 は図 1 9 と異なり、第 1 のプログラム 1 1 0 にデータ送信要求生成・通知処理部 1 2 3 を含み、第 1 のプログラム 1 1 0 にデータ送信要求生成処理部 1 2 2 を含まない。

【 0 0 6 9 】

データ送信要求生成・通知処理部 1 2 3 は、データ送信要求生成処理部 1 2 2 と異なり、送信部 1 1 7 を起動するとき、データ送信要求と通知要求を組みにして送信部に渡す。ただし、本発明を実施するためには、データ送信要求と通知要求は必ずしも組みにする必要はない。本発明の第 7 の実施の形態では CPU の負荷の削減効果が大きく、RDMA-Read をサポートしていないものにも適用が可能である。また、送信側の通知によるため受信側の負担は小さい。

【 0 0 7 0 】

次に、図 3 0 を用いて、第 8 の本発明の実施の形態と第 4 の本発明の実施の形態との違いを説明する。

図 3 0 は、第 8 の本発明の実施の形態の全体構成図である。図 3 0 は図 2 3 と異なり、第 1 のプログラム 1 1 0 にデータ送信要求生成・通知処理部 1 2 3 を含み、第 1 のプログラム 1 1 0 にデータ送信要求生成処理部 1 2 2 を含まない。本

発明の第 8 の実施の形態では読み取りが同一方向でも適用でき、RDMA-Read をサポートしていないものにも適用出来る。また、送信側の通知によるため受信側の負荷は小さい。

【0071】

次に、図 31 を用いて、第 9 の本発明の実施の形態を説明する。

図 31 は、第 9 の本発明の実施の形態の全体構成図である。第 1 の計算機ノード 10 は基幹系システム 510 を含み、第 2 の計算機ノード 20 は情報系システム 20 を含む。第 1 のプログラム 110 は、第 9 の本発明の実施の形態ではオンライン・トランザクション・プロセッシング (OLTP) である。データレコード 153 は第 9 の本発明の実施の形態では、OLTP 110 が処理過程を保存するために出力するジャーナルである。第 2 のプログラム 210 は、第 9 の本発明の実施の形態ではデータベース管理システム (DBMS) である。

【0072】

第 9 の本発明の実施の形態では、第 1 の本発明の実施の形態から、第 8 の本発明の実施の形態に示した処理内容の具体的応用を示すもので、これらのいずれか一つと同様の処理を行って、OLTP 110 が出力するジャーナル 153 を情報系システム 520 に転送する。

【0073】

【発明の効果】

本発明により、データを送信するプログラムと受信するプログラムとの同期オーバーヘッドが低減する。

【図面の簡単な説明】

【図 1】 本発明の第 1 の実施の形態の全体構成図。

【図 2】 第 1 のプログラム 110 の処理を示すフローチャート。

【図 3】 第 2 のプログラム 210 の処理を示すフローチャート。

【図 4】 識別情報出力処理 112 の処理を示すフローチャート。

【図 5】 格納処理 113 を示すフローチャート。

【図 6】 データレコードテーブル 151 を示す図。

【図 7】 データレコードテーブル 251 を示す図。

【図 8】 データレコードテーブル 1 5 1 を示す図。

【図 9】 同方向に読み出しと書き込みを行なったときのデータレコードテーブルの状態を説明する図 (1)。

【図 1 0】 同方向に読み出しと書き込みを行なったときのデータレコードテーブルの状態を説明する図 (2)。

【図 1 1】 データレコードテーブル 2 5 1 を示す図。

【図 1 2】 データレコード参照処理 2 2 1 の内容を示すフローチャート。

【図 1 3】 第 2 の本発明の実施の形態の全体構成図。

【図 1 4】 第 1 のプログラム 1 1 0 の処理を示すフローチャート。

【図 1 5】 格納処理 1 1 3 の内容を示すフローチャート。

【図 1 6】 データレコードテーブル 1 5 1 を示す図。

【図 1 7】 データレコードテーブル 1 5 1 を示す図。

【図 1 8】 データレコード参照処理 2 2 1 の内容を示すフローチャート。

【図 1 9】 第 3 の本発明の実施の形態の全体構成図。

【図 2 0】 第 1 のプログラム 1 1 0 の処理を示すフローチャート。

【図 2 1】 データレコードテーブル 2 5 1 を示す図。

【図 2 2】 データレコード参照処理 2 2 1 の内容を示すフローチャート。

【図 2 3】 第 4 の本発明の実施の形態の全体構成図。

【図 2 4】 第 1 のプログラム 1 1 0 の処理を示すフローチャート。

【図 2 5】 第 5 の本発明の実施の形態の全体構成図。

【図 2 6】 第 1 のプログラム 1 1 0 の処理を示すフローチャート。

【図 2 7】 第 2 のプログラム 2 1 0 の処理を示すフローチャート。

【図 2 8】 第 6 の本発明の実施の形態の全体構成図。

【図 2 9】 第 7 の本発明の実施の形態の全体構成図。

【図 3 0】 第 8 の本発明の実施の形態の全体構成図。

【図 3 1】 第 9 の本発明の実施の形態の全体構成図。

【符号の説明】

1 0 : 第 1 の計算機ノード

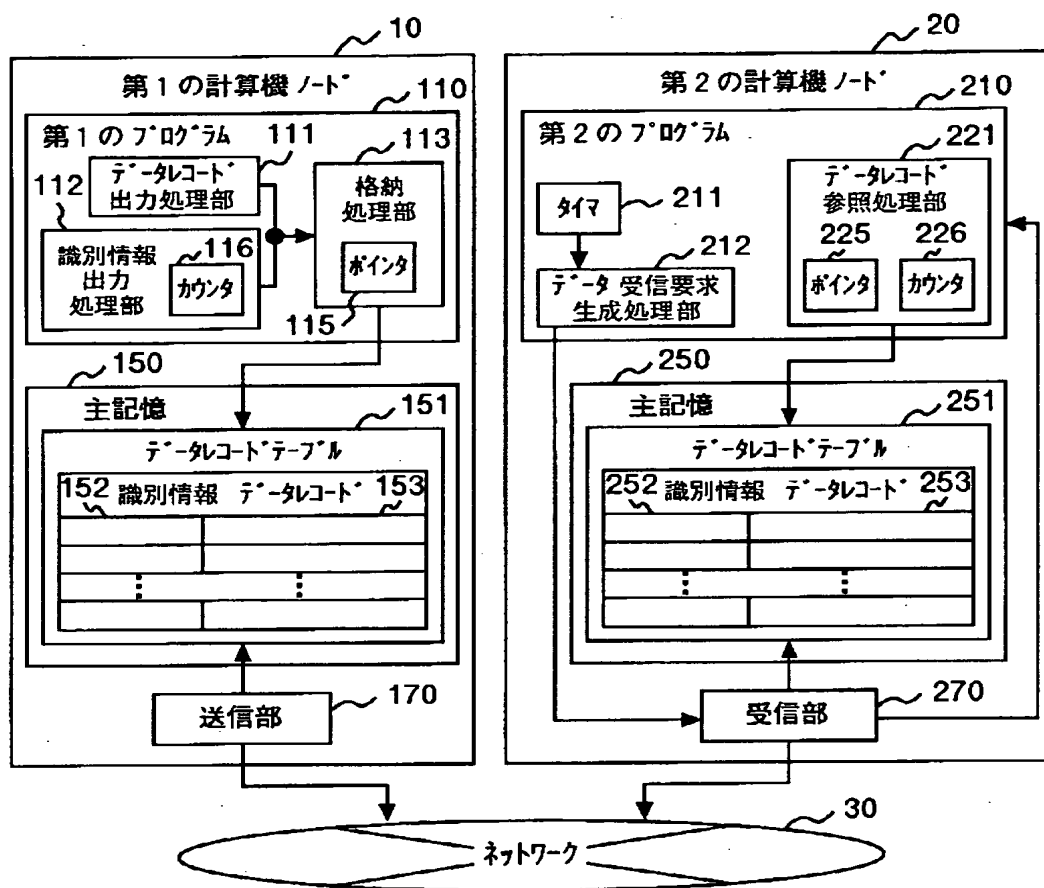
2 0 : 第 2 の計算機ノード

1 1 0 : 第 1 の プ ロ グ ラ ム
1 1 1 : デ ー タ レ コ ー ド 出 力 処 理 部
1 1 2 : 識 別 情 報 出 力 処 理 部
1 1 3 : 格 納 処 理 部
1 1 5 : ポ イ ン タ
1 1 6 : カ ウ ン タ
1 5 0 : 主 記 憶
1 5 1 : デ ー タ レ コ ー ド テ ー ブ ル
1 5 2 : 識 別 情 報
1 5 3 : デ ー タ レ コ ー ド
1 7 0 : 送 信 部
2 1 0 : 第 2 の プ ロ グ ラ ム
2 1 1 : タ イ マ
2 1 2 : デ ー タ 受 信 要 求 生 成 処 理 部
2 2 1 : デ ー タ レ コ ー ド 参 照 処 理 部
2 2 5 : ポ イ ン タ
2 2 6 : カ ウ ン タ
2 5 0 : 主 記 憶
2 5 1 : デ ー タ レ コ ー ド テ ー ブ ル
2 5 2 : 識 別 情 報
2 5 3 : デ ー タ レ コ ー ド
2 7 0 : 受 信 部
3 0 : ネ ッ ト ワ ー ク

【書類名】 図面

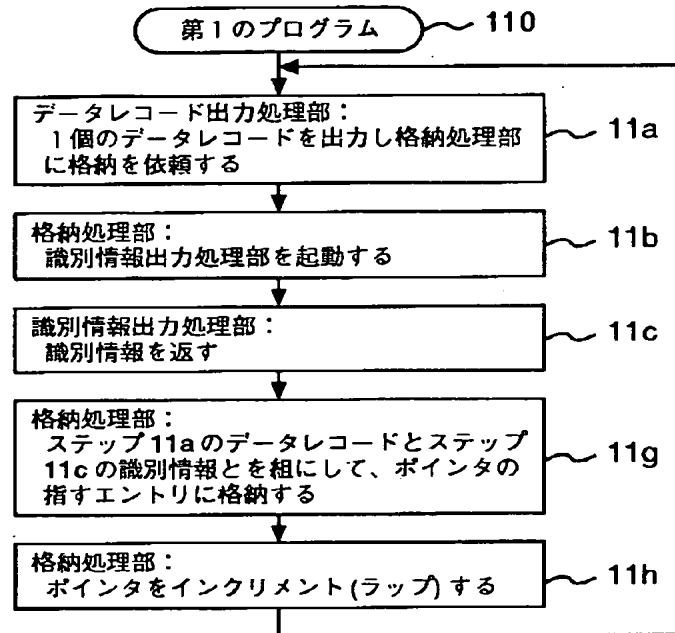
【図 1】

図 1



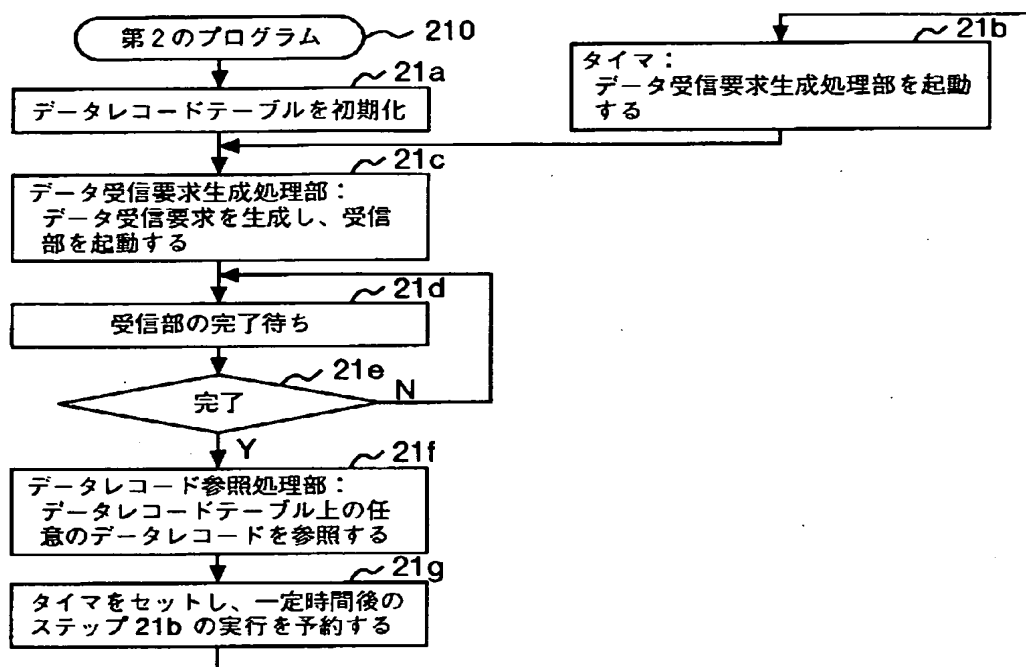
【図 2】

図 2



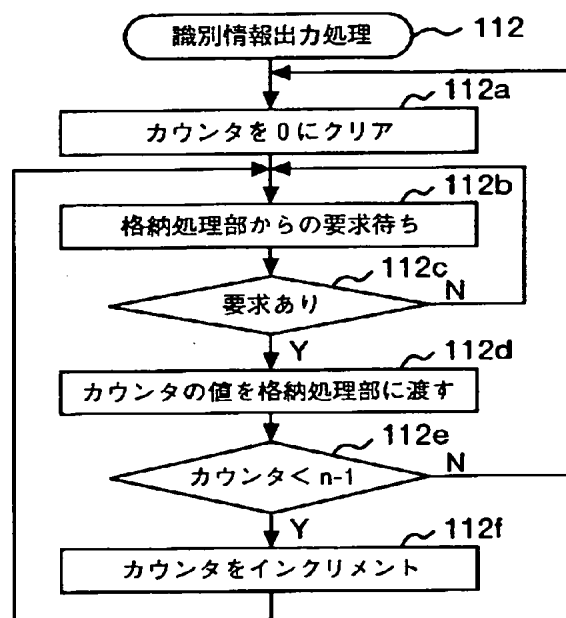
【図 3】

図 3



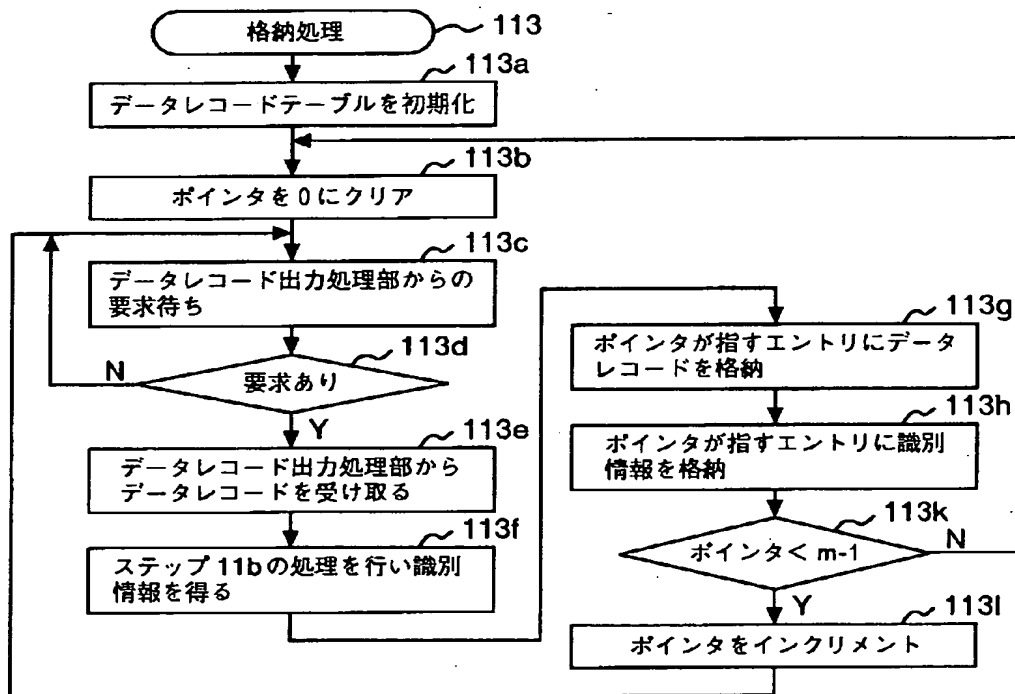
【図 4】

図 4



【図 5】

図 5



【図 6】

図 6

151

データレコードテーブル		識別情報	データレコード	
エントリ: m-1 →	152.m-1 ~	n-1	空のデータレコード	~153.m-1
エントリ: m-2 →	152.m-2 ~	m-3	空のデータレコード	~153.m-2
エントリ: m-3 →	152.m-3 ~	m-4	空のデータレコード	~153.m-3
		⋮	⋮	
エントリ: 2 →	152.2 ~	1	空のデータレコード	~153.2
エントリ: 1 →	152.1 ~	0	空のデータレコード	~153.1
エントリ: 0 →	152.0 ~	-1	空のデータレコード	~153.0

【図 7】

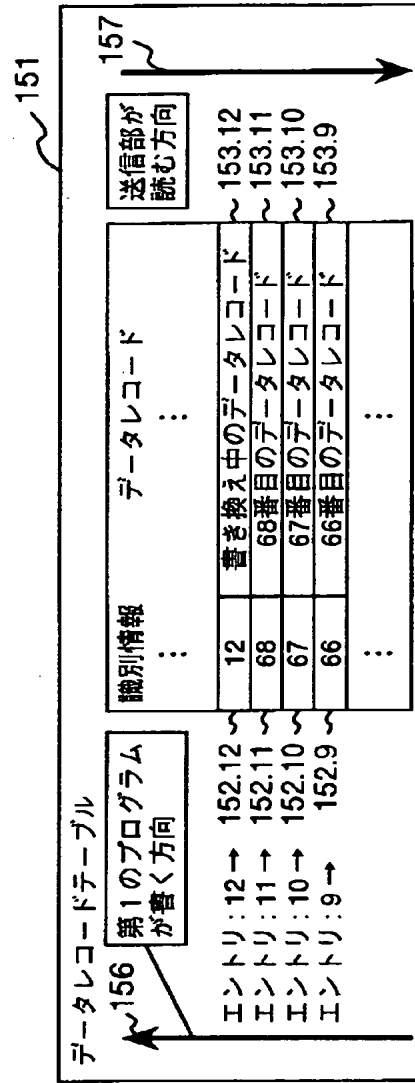
図 7

251

データレコードテーブル		
		識別情報 データレコード
エントリ: 1-1 →	252.1-1 ~	n-1 空のデータレコード ~ 253.1-1
エントリ: 1-2 →	252.1-2 ~	1-3 空のデータレコード ~ 253.1-2
エントリ: 1-3 →	252.1-3 ~	1-4 空のデータレコード ~ 253.1-3
		⋮ ⋮
エントリ: 2 →	252.2 ~	1 空のデータレコード ~ 253.2
エントリ: 1 →	252.1 ~	0 空のデータレコード ~ 253.1
エントリ: 0 →	252.0 ~	-1 空のデータレコード ~ 253.0

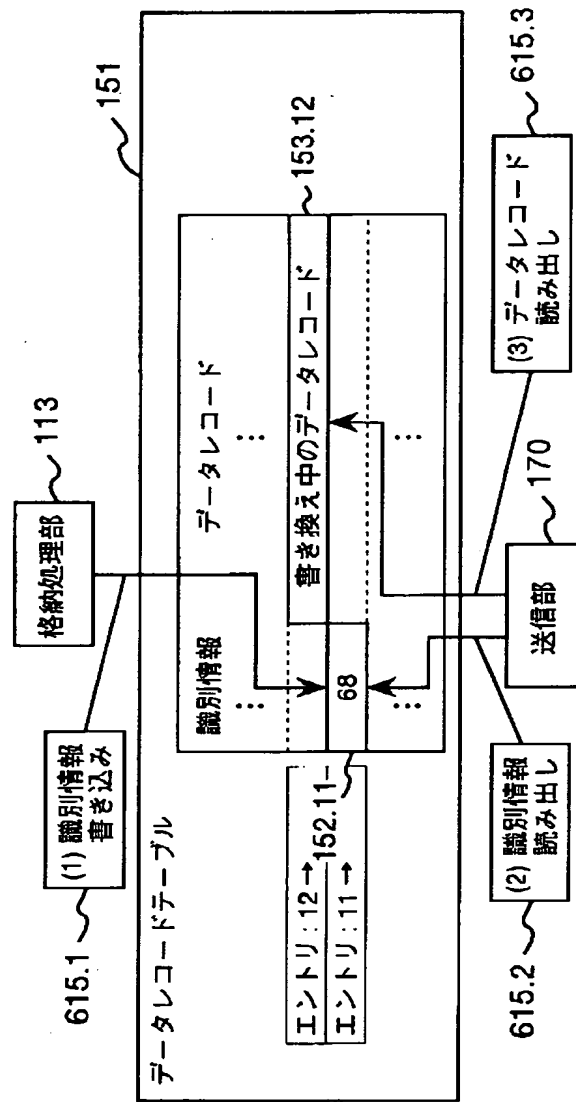
【図 8】

図 8

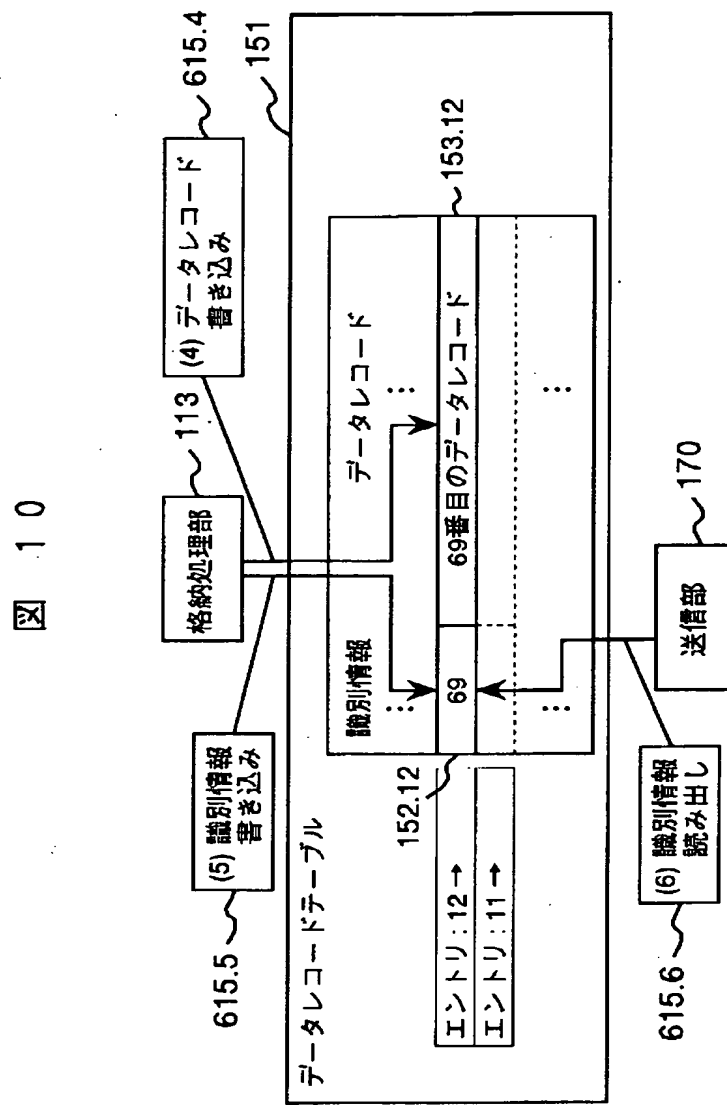


【図9】

図 9



【図 10】



【図 11】

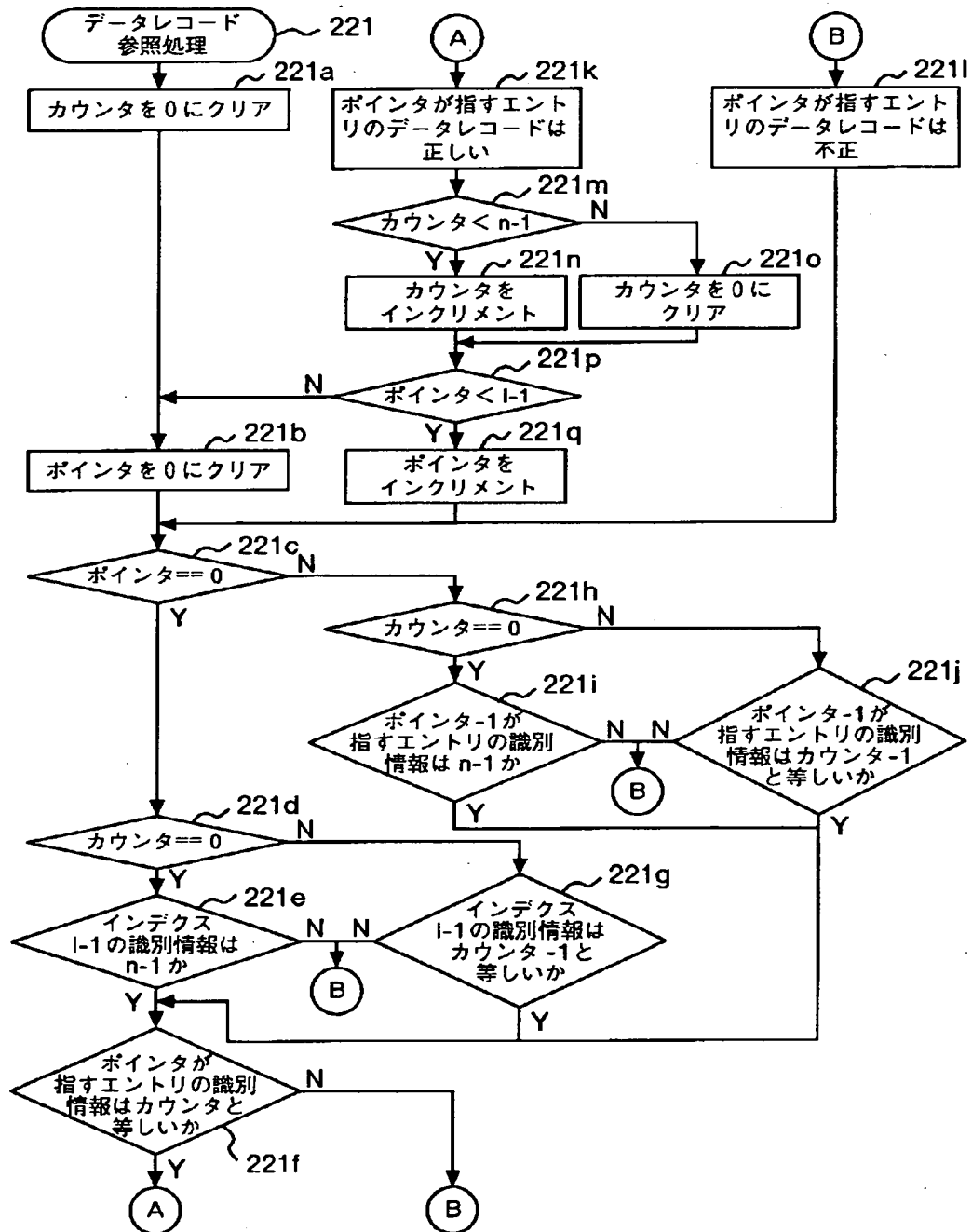
図 11

251

データレコードテーブル		識別情報	データレコード
		⋮	⋮
エントリ: 12 →	252.12 ~	12	書き換え中のデータレコード ~ 253.12
エントリ: 11 →	252.11 ~	68	68番目のデータレコード ~ 253.11
エントリ: 10 →	252.10 ~	67	67番目のデータレコード ~ 253.10
エントリ: 9 →	252.9 ~	66	66番目のデータレコード ~ 253.9
		⋮	⋮

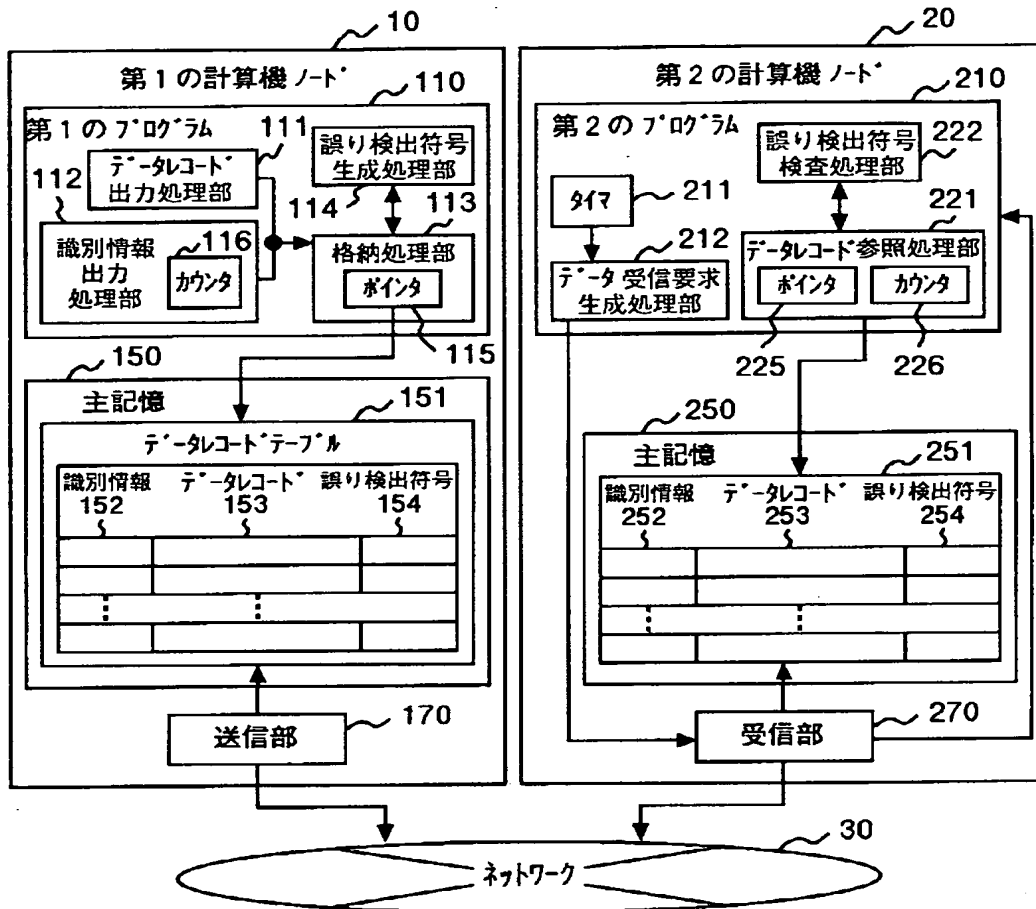
【図 1 2】

図 1 2



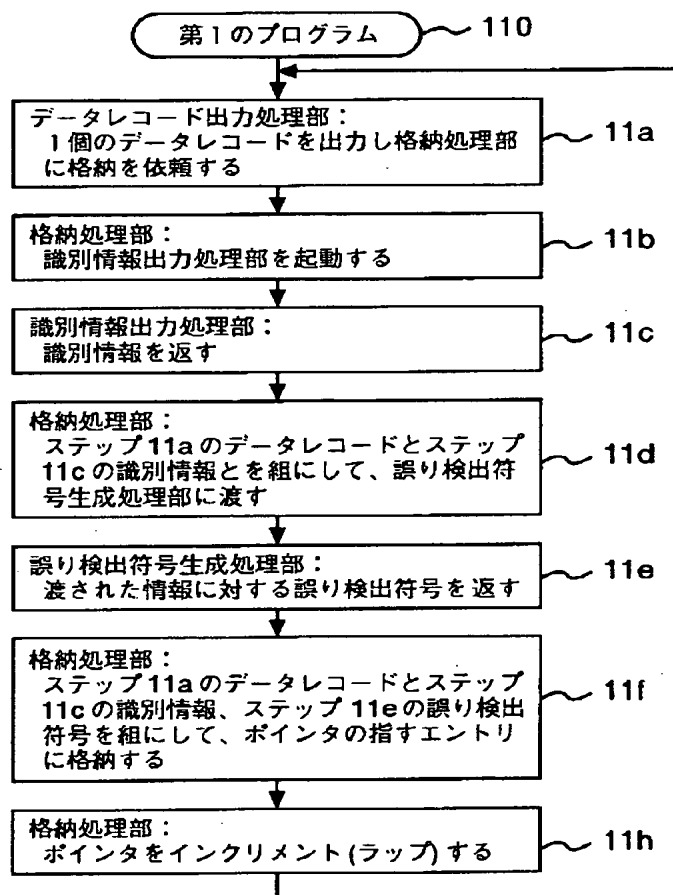
【図 1 3】

図 1 3



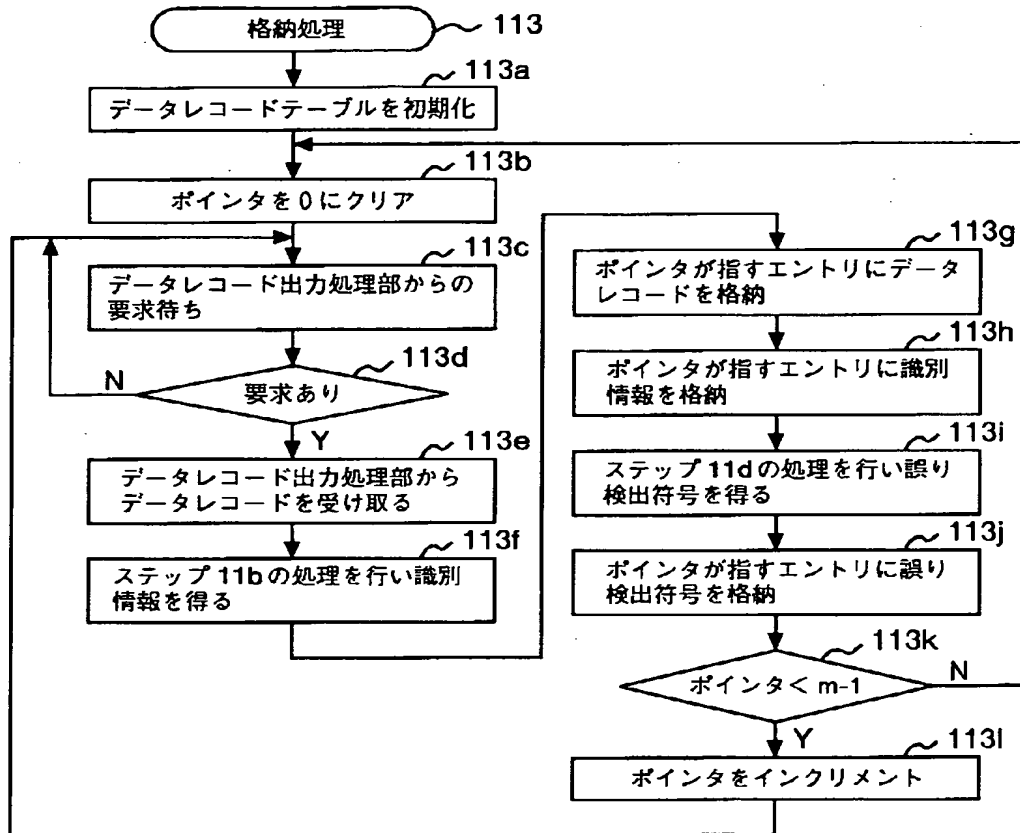
【図 14】

図 14



【図 1 5】

図 1 5



【図 1 6】

図 1 6

151

データレコードテーブル		153		
		識別情報	データレコード	誤り訂正符号
エントリ: m-1 → 152.m-1 ~		n-1	空のデータレコード	不正な符号 ~ 154.m-1
エントリ: m-2 → 152.m-2 ~		m-3	空のデータレコード	不正な符号 ~ 154.m-2
エントリ: m-3 → 152.m-3 ~		m-4	空のデータレコード	不正な符号 ~ 154.m-3
		⋮	⋮	⋮
エントリ: 2 → 152.2 ~		1	空のデータレコード	不正な符号 ~ 154.2
エントリ: 1 → 152.1 ~		0	空のデータレコード	不正な符号 ~ 154.1
エントリ: 0 → 152.0 ~		-1	空のデータレコード	不正な符号 ~ 154.0

【図 1 7】

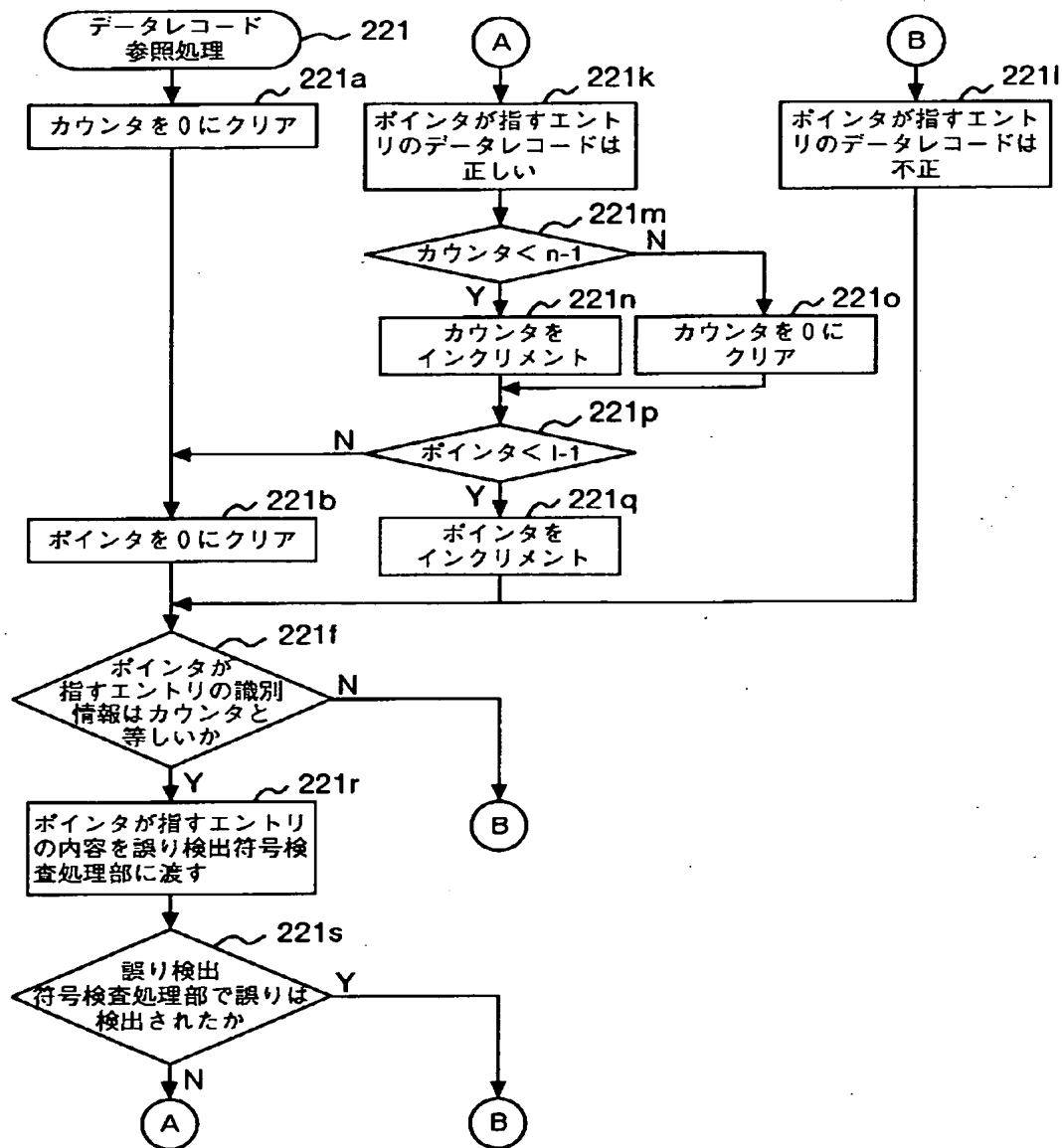
図 1 7

151

データレコードテーブル		153		
		識別情報	データレコード	誤り検出符号
		⋮	⋮	⋮
エントリ: 12 → 152.12 ~		12	書き換え中のデータレコード	不正な符号 ~ 154.12
エントリ: 11 → 152.11 ~		68	68番目のデータレコード	正しい符号 ~ 154.11
エントリ: 10 → 152.10 ~		67	67番目のデータレコード	正しい符号 ~ 154.10
エントリ: 9 → 152.9 ~		66	66番目のデータレコード	正しい符号 ~ 154.9
		⋮	⋮	⋮

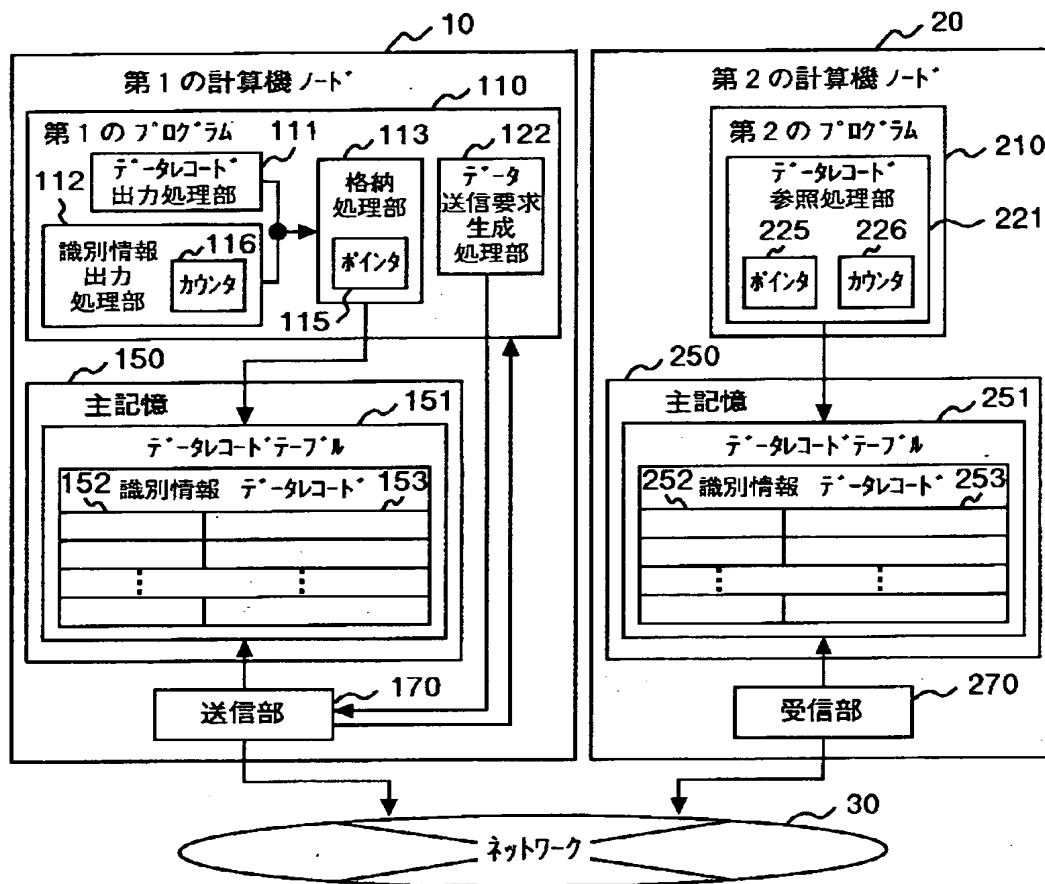
【図 18】

図 18



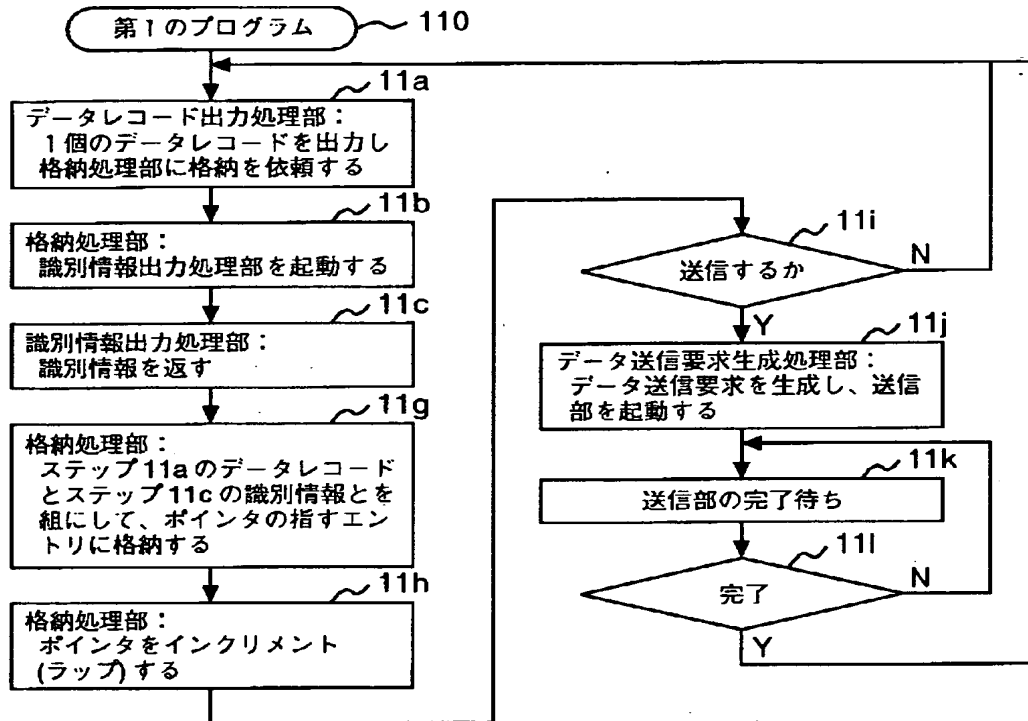
【図19】

図 19



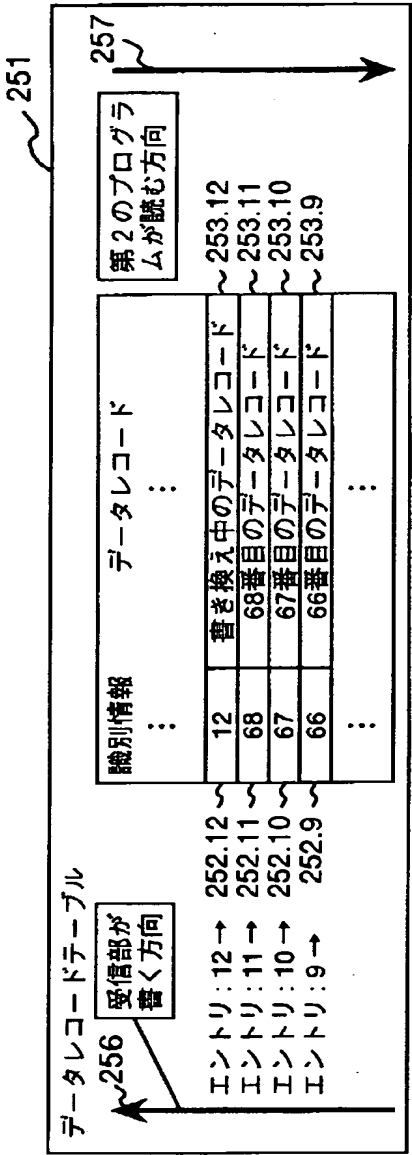
【図 2 0】

図 2 0



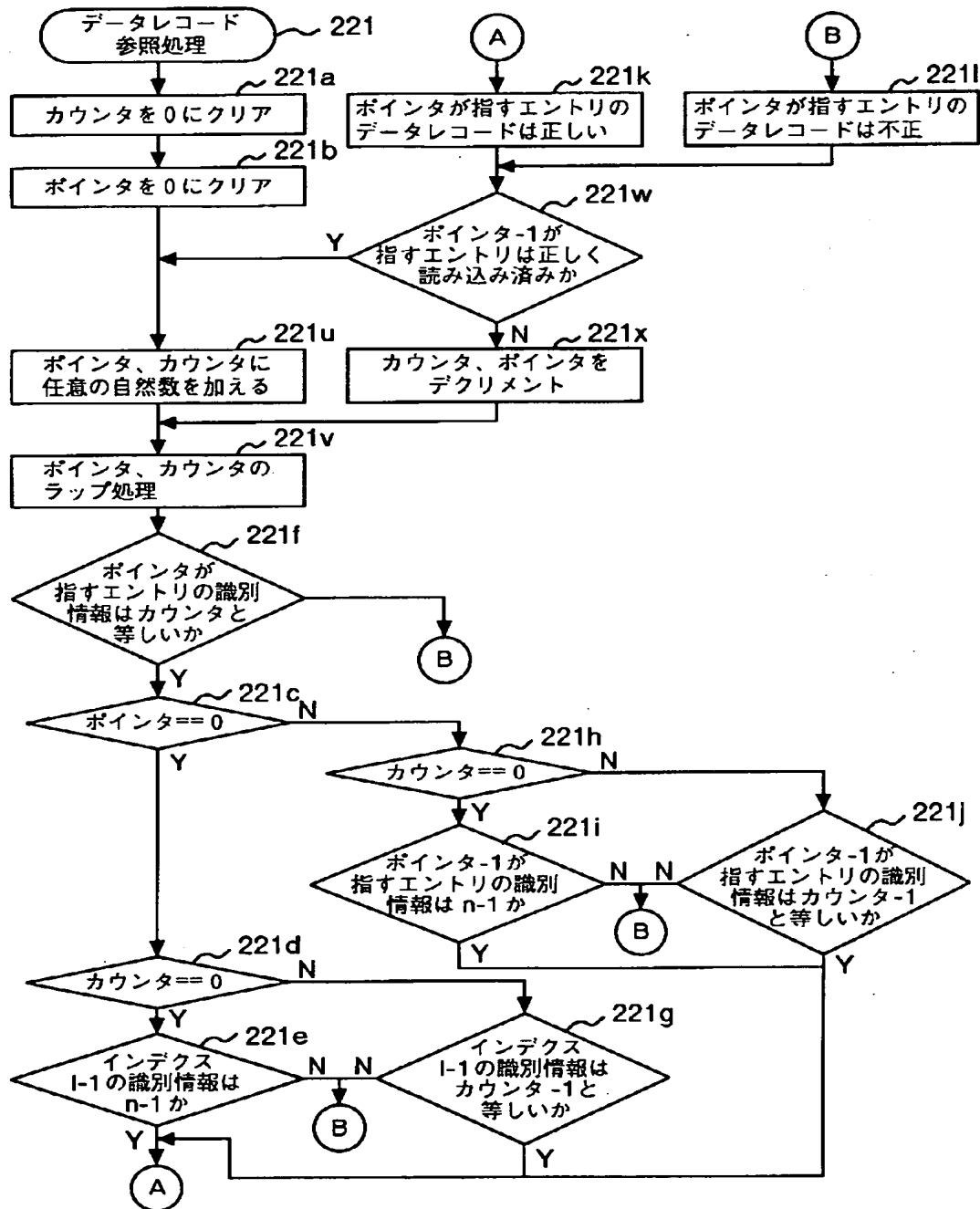
【図 2 1】

図 2 1



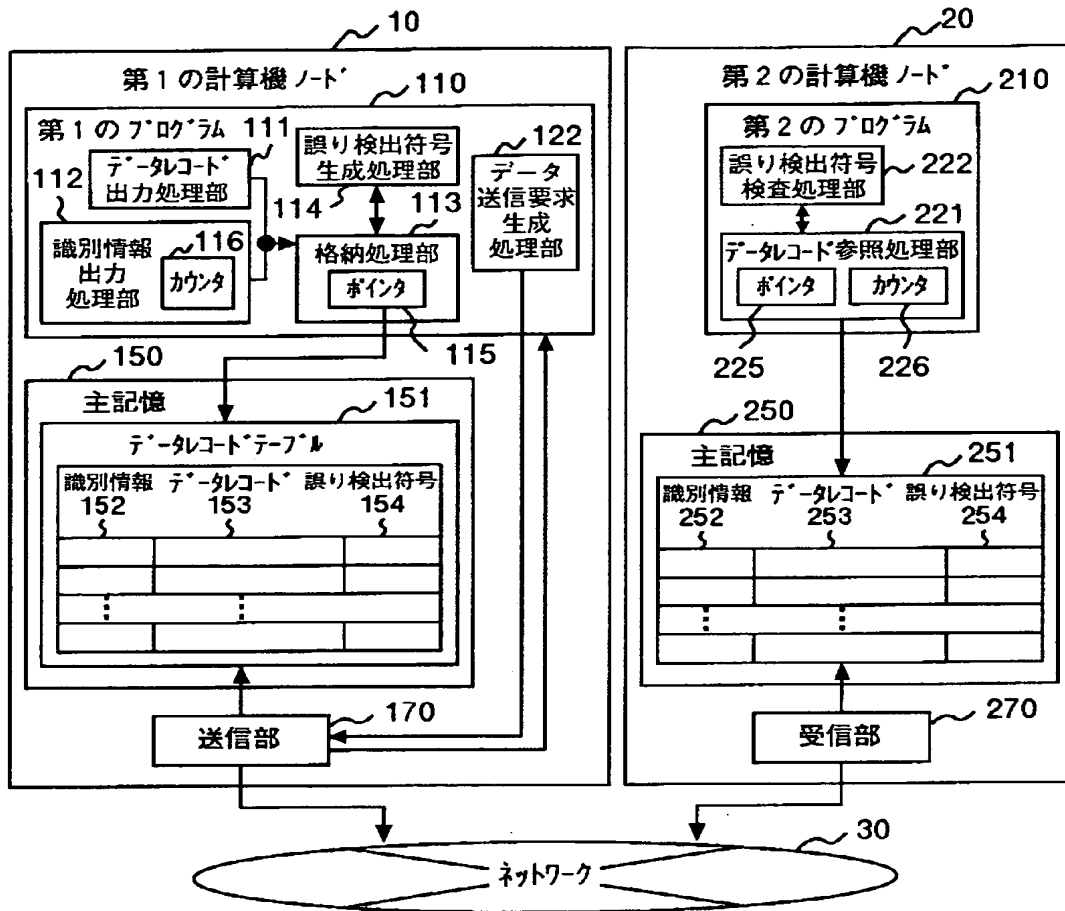
【図 22】

図 22



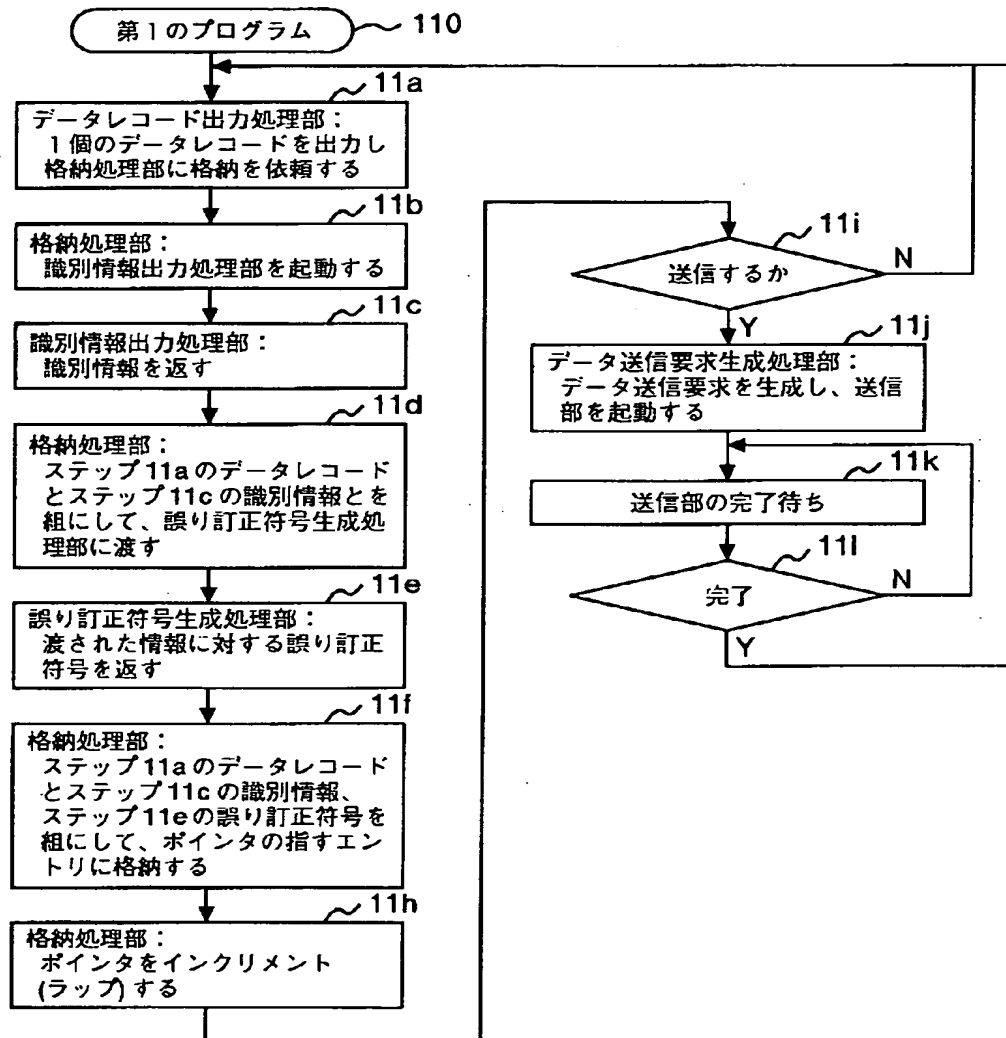
【図 23】

図 23



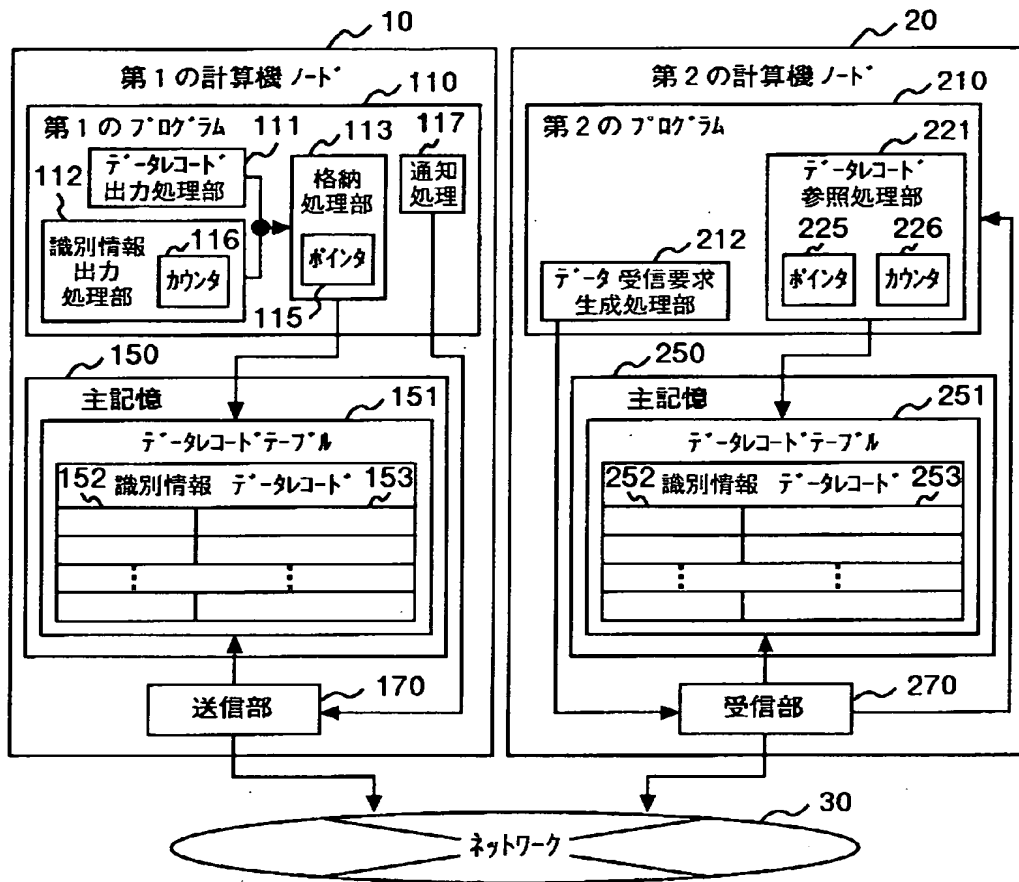
【図 24】

図 24



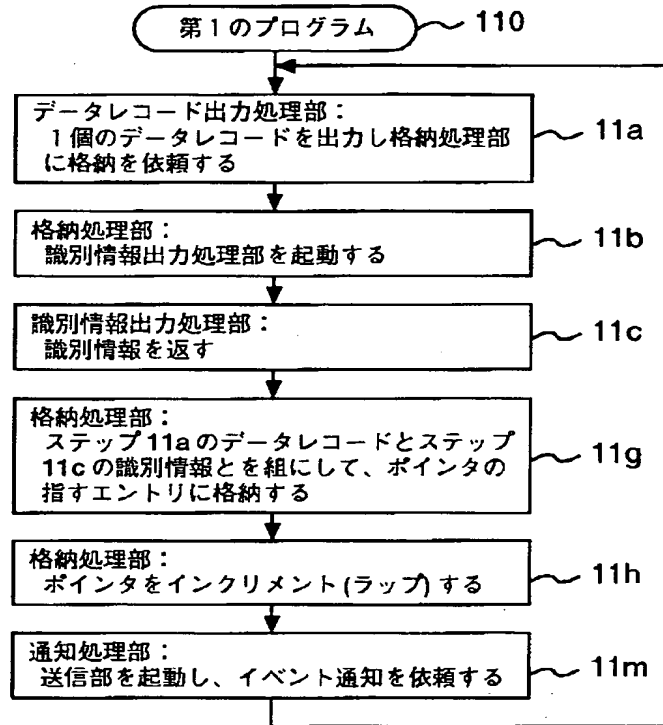
【図 2 5】

図 2 5



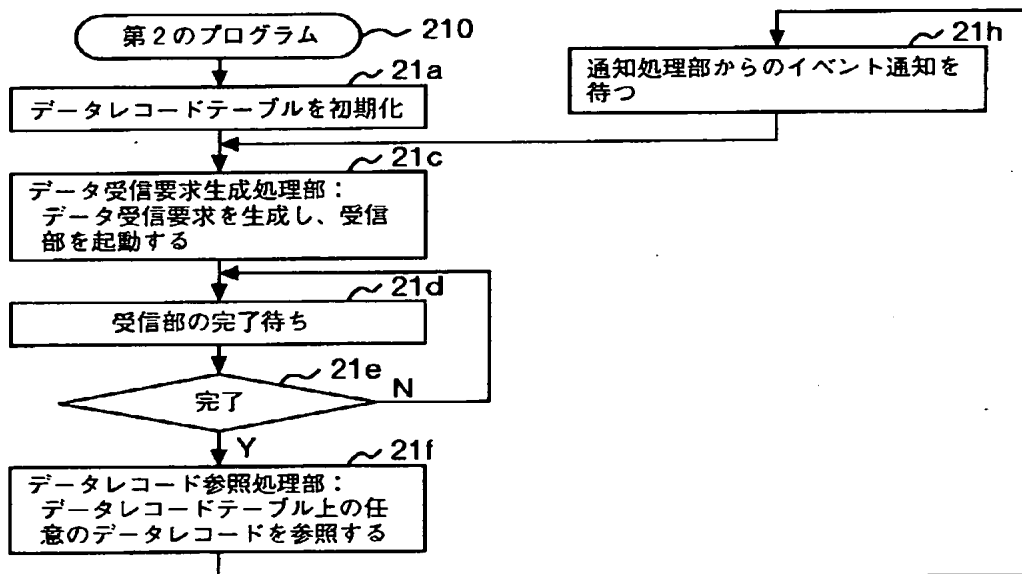
【図 2 6】

図 2 6



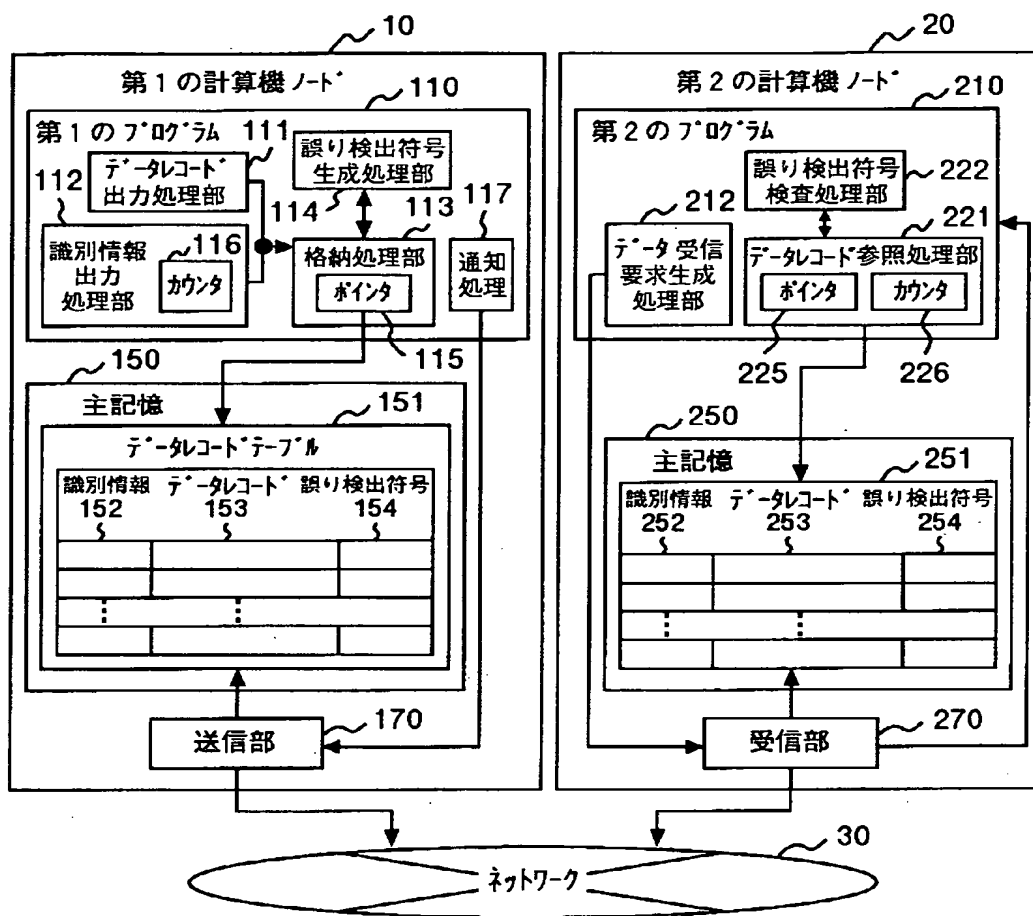
【図 2 7】

図 2 7



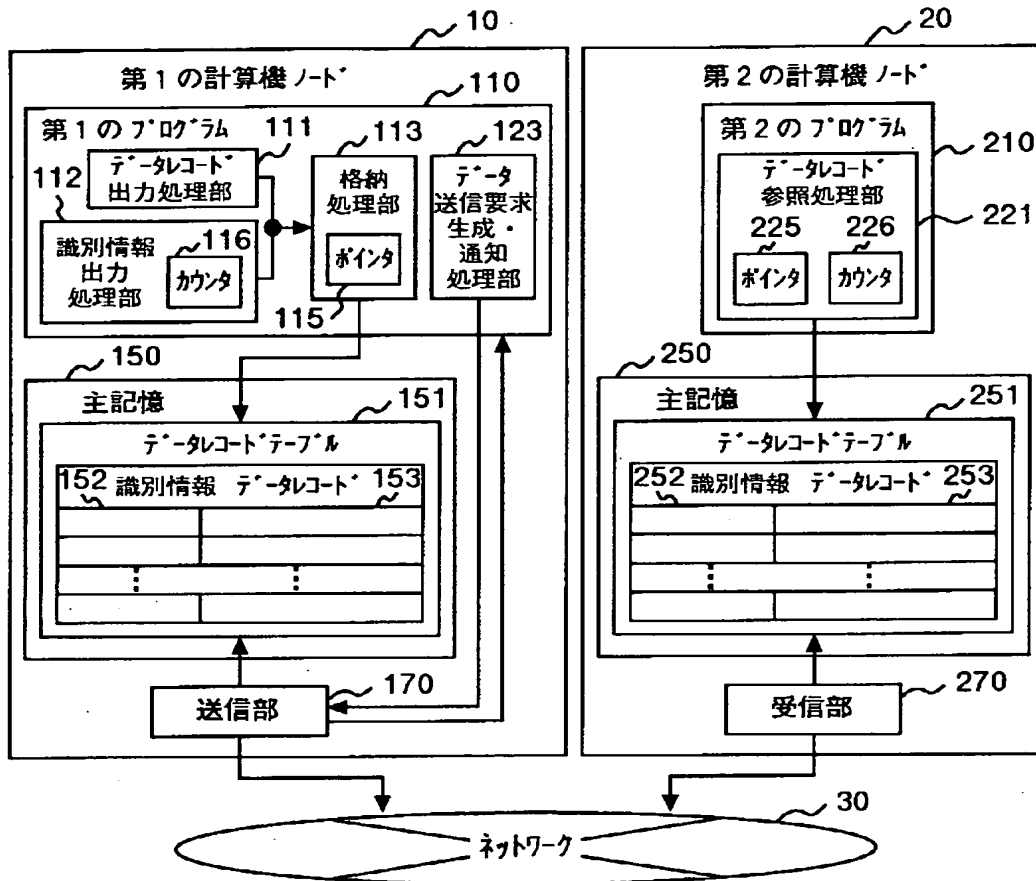
【図 28】

図 28



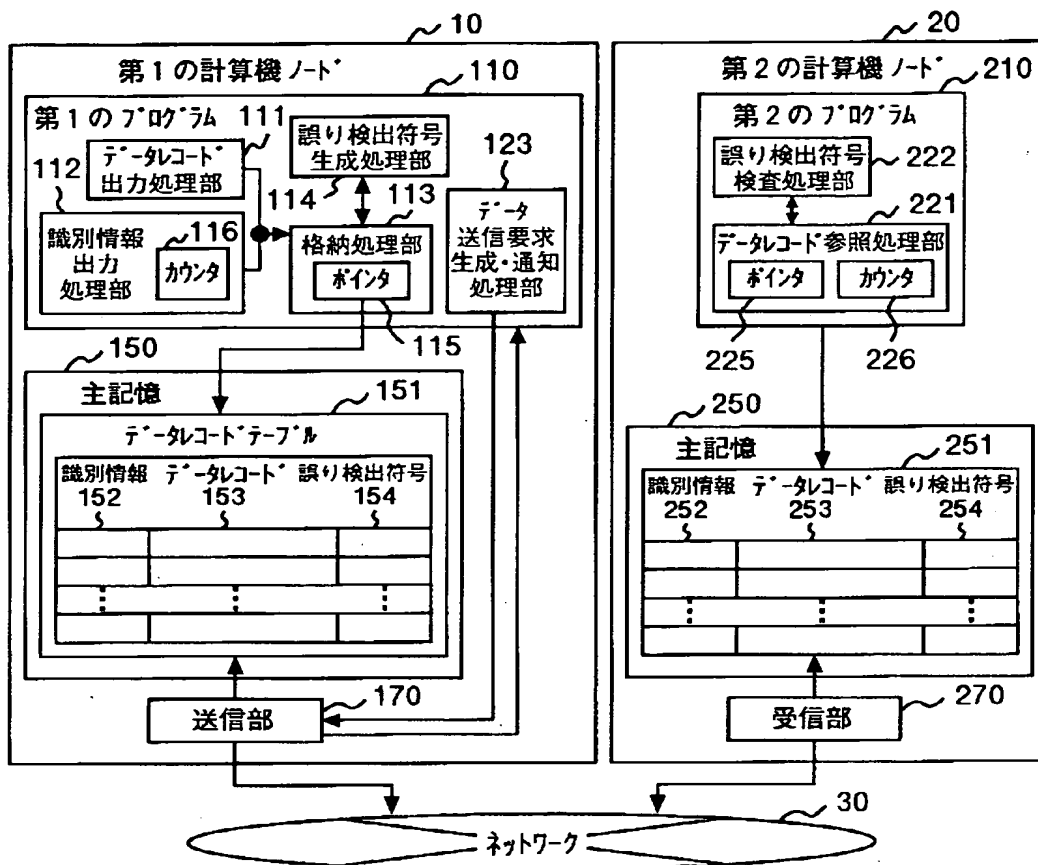
【図 29】

図 29



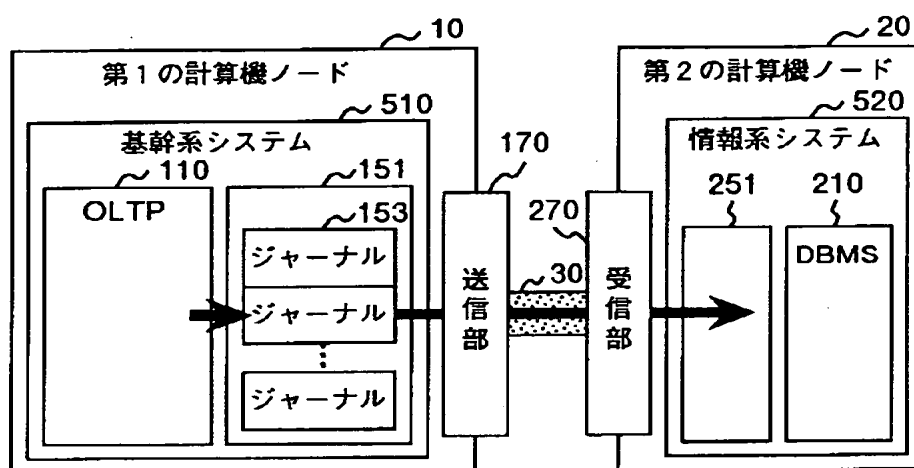
【図 30】

図 30



【図 3 1】

図 3 1



【書類名】 要約書

【要約】

【課題】従来の2つの計算機ノード間のデータ受け渡し方法では、データ転送を行うプログラム間の待ち合わせのオーバーヘッドが大きい。

【解決手段】データを送信する第1のプログラムが、任意の時間間隔でデータを主記憶の領域に格納する。データを受信する第2のプログラムは、任意の時間間隔で上記領域をRDMAを用いて、参照する。または、データを受信する第2のプログラムは、任意の時間間隔で主記憶の領域を参照する。データを送信する第1のプログラムは、任意の時間間隔でRDMAを用いてデータを上記領域に格納する。さらに、書き込みと読み出しのすれ違いを検出するため、各レコードに識別子を付け、識別子とレコード本体のアクセス順序を書き込みと読み出しで逆にする。あるいは各レコードに識別子と誤り検出符号を付加する。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 1 0 8]

1. 変更年月日	1 9 9 0 年 8 月 3 1 日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所